

# **Development and Application of a Description-based Interface for 3D Object Reconstruction**

---

Kai Wu

January 2, 2018

University of British Columbia  
[kaywu@ece.ubc.ca](mailto:kaywu@ece.ubc.ca)

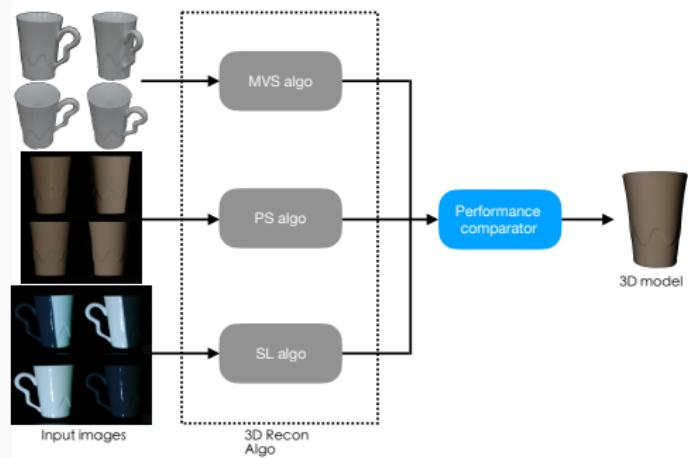
# Table of contents

1. Introduction
2. Related Work
3. Development of Interface
4. Evaluation of interface
5. Conclusions

# Introduction

---

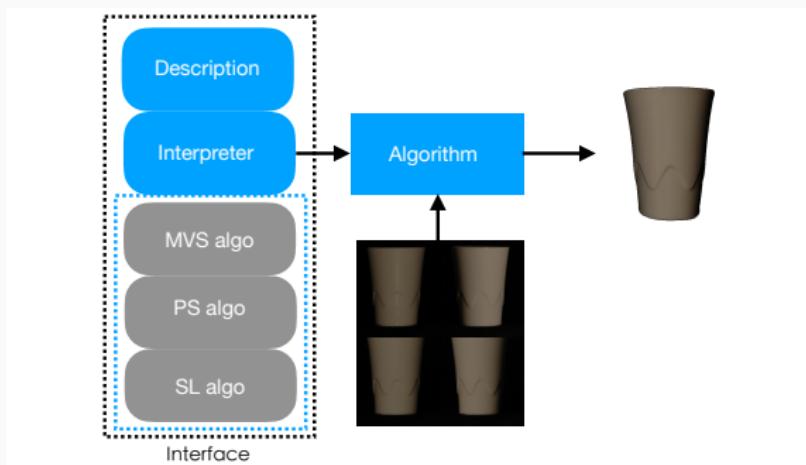
# Motivation: traditional 3D reconstruction



## Challenges

- Algorithms: vision knowledge required;
- Parameters: not interpretable, meaningful, or conceptually estimable;
- Approach: *try-and-error*.

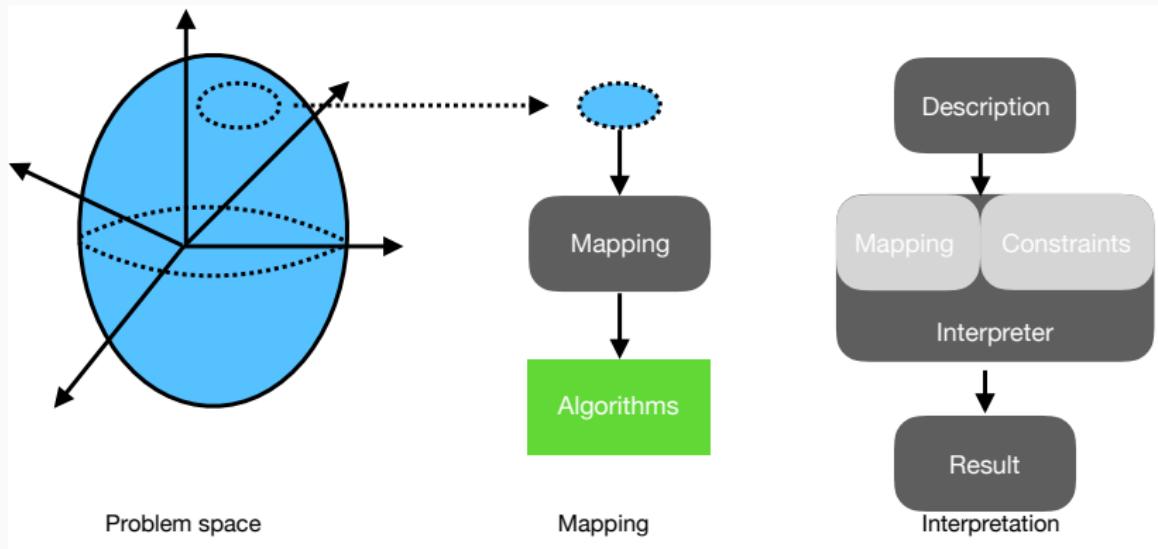
# Motivation: interface to 3D reconstruction



## Strengths

- Algorithms: description of appearance, no vision background needed, embedding new algorithms is easy;
- Parameters: property parameters are perceptually interpretable& meaningful;
- Approach: choose an algorithm based on mapping.

# Overview



## Contribution

Development of an interface for 3D reconstruction problem, which hides algorithmic details and allows users to describe conditions surrounding the problem. This description can be interpreted so that an appropriate algorithm is chosen to achieve a successful reconstruction result.

## Related Work

---

## Related Work: softwares

Some notable open source general vision libraries and softwares:

### General vision libraries

- Example: OpenCV, VXL, VLFeat, and so on
- Problem: provide APIs for vision routines

### 3D vision softwares

- Example: PMVS; Bundler, VisualSfM, TheiaSfM; Poisson Recon;
- Problem: cater to specific objects, not applicable for textureless surface

### Challenges

1. Not that we don't have enough tools, but the barrier to take advantage of these tools is high.

# Related Work: algorithms

## Shape from Stereo

- Example: Multi-View Stereo, Structured Light
- Problem: Texture, reflectance

## Shape from Intensity

- Example: Shape from Shading, Photometric Stereo
- Problem: Lightness, shape

## Shape from Silhouette

- Example: Visual Hull, Space Carving
- Problem: Shape, reflectance

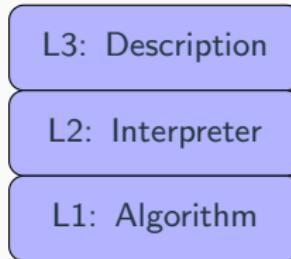
## Challenges

1. Few algorithm works for objects with diverse range of properties;
2. The range of problem conditions under which an algorithm works is not known a priori.

## **Development of Interface**

---

# Overview



**Figure 1:** 3-layer interface to 3D reconstruction.

## Description

1. define problem space;
2. describe problem condition.

**Interpreter: translate description to an appropriate algorithm.**

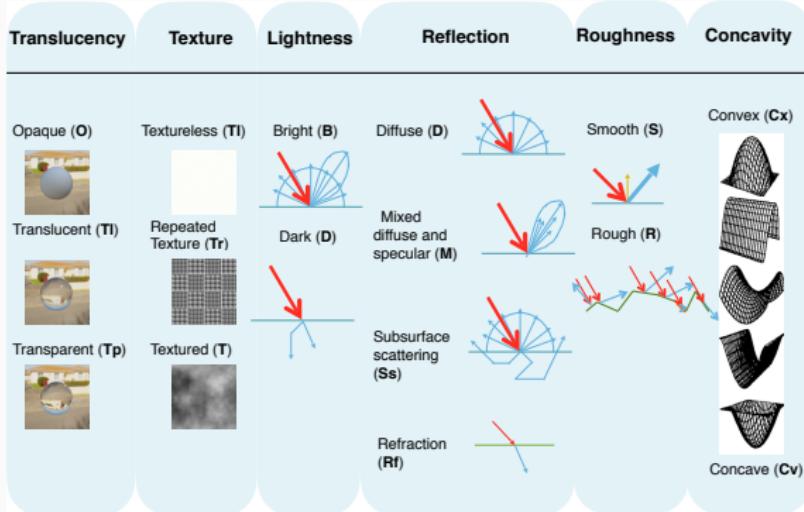
Mapping: discover the relation between problem space and algorithm.

## Algorithm

Embed algorithms into the interface

# Problem space

- *algorithm-centered* approach categorizes algorithms based on algorithmic details, as discussed in **Related Work**;
- *object-centered* taxonomy categorizes algorithms based on the problem conditions that the algorithm can reliably work under.



# Problem space: four problem conditions

Assumptions:

- Active methods require high surface albedo (bright), in order to demonstrate the effectiveness of these methods, we focus on bright surfaces only.
- Diffuse is caused solely by surface roughness since sub-surface scattering is ignored.

| Condition | Texture          | Lightness    | Reflection | Roughness  | Label       |           |            |           |          |
|-----------|------------------|--------------|------------|------------|-------------|-----------|------------|-----------|----------|
| 1         | Textureless (Tl) | Textured (T) | Dark (D)   | Bright (B) | Diffuse (D) | Mixed (M) | Smooth (S) | Rough (R) | TI-B-D-R |
| 2         | Yes              |              |            | Yes        | Yes         |           | Yes        |           | TI-B-M-S |
| 3         |                  | Yes          |            | Yes        | Yes         |           |            | Yes       | T-B-D-R  |
| 4         |                  | Yes          |            | Yes        |             | Yes       | Yes        |           | T-B-M-S  |

## Description: model and representations

| Model       | Representation              |
|-------------|-----------------------------|
| Texture     | <i>Texture randomness</i>   |
| Lightness   | <i>Diffuse albedo</i>       |
| Specularity | <i>Specular reflectance</i> |
| Roughness   | <i>SD of facet slopes</i>   |

**Table 1:** Representations of the 3D reconstruction problem.

## Description: expression

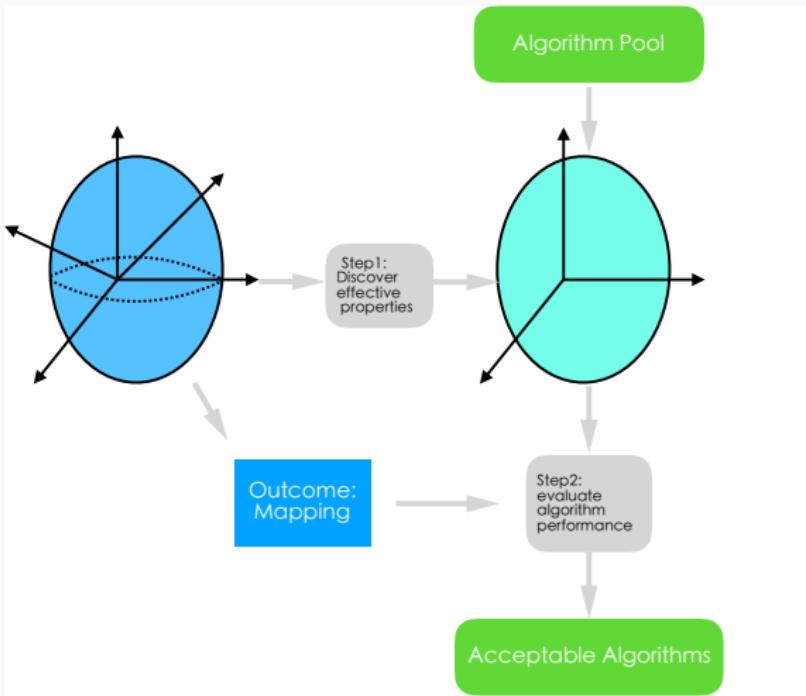
We use three discrete scales to parameterize these properties: *low* (0.2), *medium* (0.5), and *high* (0.8).

| <b>Prob cond</b> | Texture | Albedo | Specular | Rough   | <b>Label</b> |
|------------------|---------|--------|----------|---------|--------------|
| 1                | low/med | high   | low/med  | high    | TI-B-D-R     |
| 2                | low/med | high   | high     | low/med | TI-B-M-S     |
| 3                | high    | high   | low/med  | high    | T-B-D-R      |
| 4                | high    | high   | high     | low/med | T-B-M-S      |

**Table 2:** Expression of the four problem conditions.

# Mapping

Investigate the problem conditions under which the algorithms can reliably work.



# Mapping: algorithms

## selected algorithms

- Patch-based Multi-View Stereo (PMVS): propagate-refinement-filtering;
- Example-based Photometric Stereo (EPS): arbitrary BRDF is a linear combination of basis BRDFs;
- Gray-coded Structured Light (GSL): encode spatial informally temporally.

## baseline methods

- Volumetric Visual Hull: carve voxels projecting outside of silhouettes;
- Linear-least squares Photometric Stereo:  $\mathbf{I} = \rho \mathbf{N} \cdot \mathbf{L}$

# Mapping: dataset

## creation of dataset

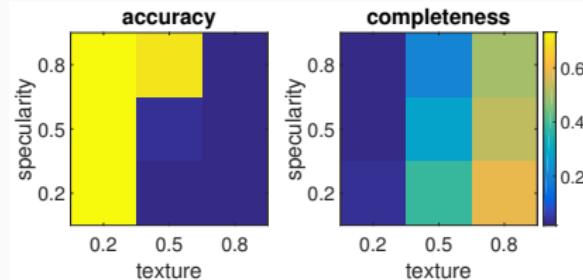
- We need a dataset contains object with varied visual and geometric properties;
- Generate the dataset using physic-based renderer.

## example of synthetic object



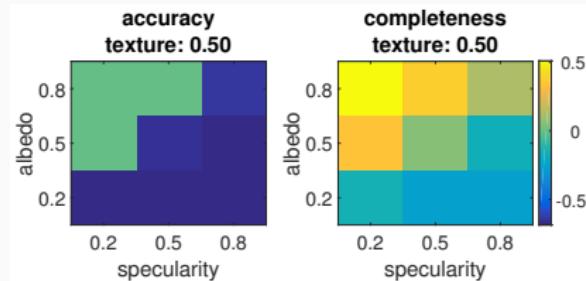
# Mapping: reduce dimensionality

Reduce problem space dimensionality by discovering properties that have an effect on algorithm performance.



# Mapping: discover mapping

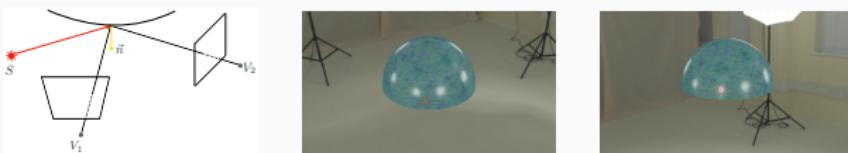
Discover the mapping from problem condition to algorithms



| Texture | Albedo | Specular | Roughness |
|---------|--------|----------|-----------|
| 0.5     | 0.5    | 0.2      | -         |
| 0.5     | 0.8    | 0.2      | -         |
| 0.5     | 0.8    | 0.5      | -         |

## Mapping: notable observations

1. PMVS can work on specular surfaces provided the surface highly textured



2. EPS and GSL fails on highly specular surfaces, and a blurred specular area leads to worse results.



(a). rough: 0.2



(b). rough: 0.5



(c). rough: 0.8

## Evaluation of interface

---

# Interpretation: evaluation methodology

## Evaluation questions

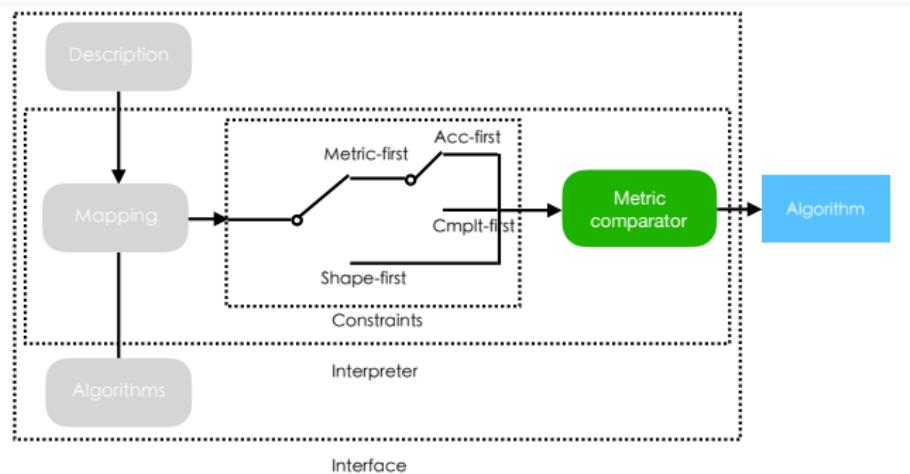
1. Interpreter returns successful reconstruction result given an accurate description of problem condition;
2. Interpreter returns a less successful reconstruction result given a less accurate description of problem condition?
3. Interpreter returns a poor reconstruction result given an inaccurate description of problem condition?

## Criteria

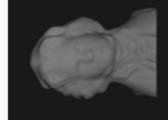
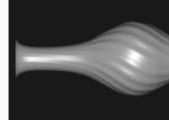
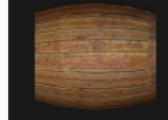
Visual comparison to results of baseline method.

# Interpretation: interpreter

An interpreter selects an appropriate algorithm based on description of problem condition and constraints.

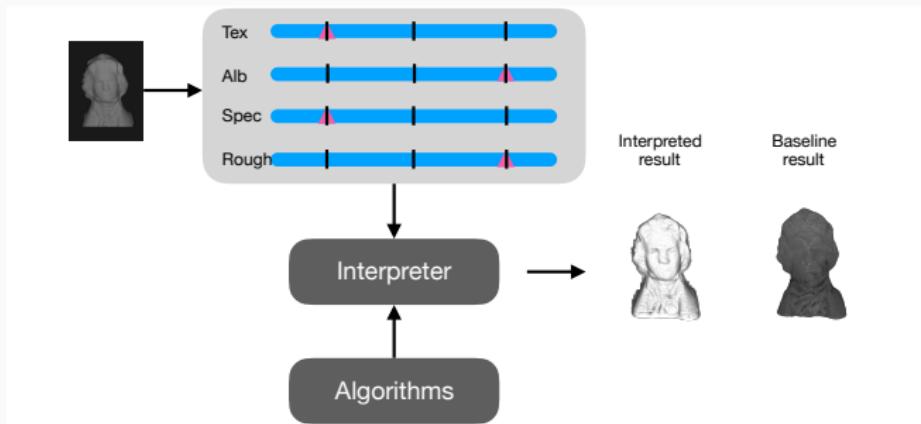


## Interpretation: dataset

| Cond#       | 1   | 2   | 3   | 4  |
|-------------|---|---|---|--|
| description | textureless<br>diffuse<br>bright  | textureless<br>mixed d/s<br>bright  | textured<br>diffuse<br>dark/bright  | textured<br>mixed d/s<br>dark/bright   |
| object      |  |  |  |  |
|             |  |  |  |  |

# Interpretation 1: demonstrative result

We use object ‘bust’ as an example:



## Comparison to baseline result

- accuracy: higher quality
- completeness: no unconstructed holes

## Interpretation 1: more results

| Algo    | GSL   | EPS   | GSL   | PMVS   |
|---------|---|---|---|--|
| Results |  |  |  |  |
|         |  |  |  |  |
| Results |  |  |  |  |
|         |  |  |  |  |

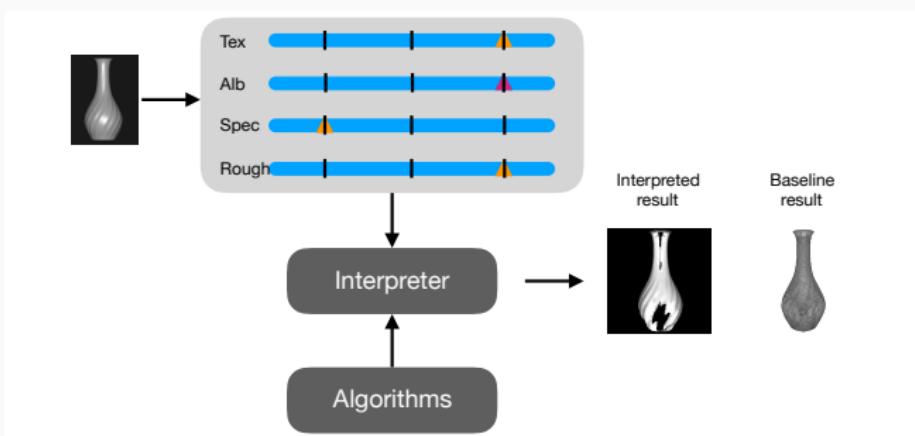
# Interpretation 1

## Summary

- We have demonstrated that it is feasible to achieve a successful reconstruction result given a description of problem condition, without knowledge of which algorithm to use;
- This could be extended to a wider range of problem conditions given a more complicated object description.

## Interpretation 2: demonstrative result

We use object 'vase' as an example:



### Comparison to baseline result

- completeness: has unconstructed holes

## Interpretation 2: more results

| Object | Descriptions and Results  |   |   |   |   |
|--------|---|---|---|---|---|
|        | $Desc_1$  | $Desc_2$  | $Desc_3$  | $Desc_4$  | Correct Desc  |
| Desc   | 02020208  | 08080208  | 08020802  | 08020202  | 02080802  |
| Algo   | • EPS   | • PMVS<br>• EPS<br>• GSL  | • BL  | • PMVS  | • EPS   |
| vase0  |  |  |  |  |  |
| cup    |  |  |  |  |  |

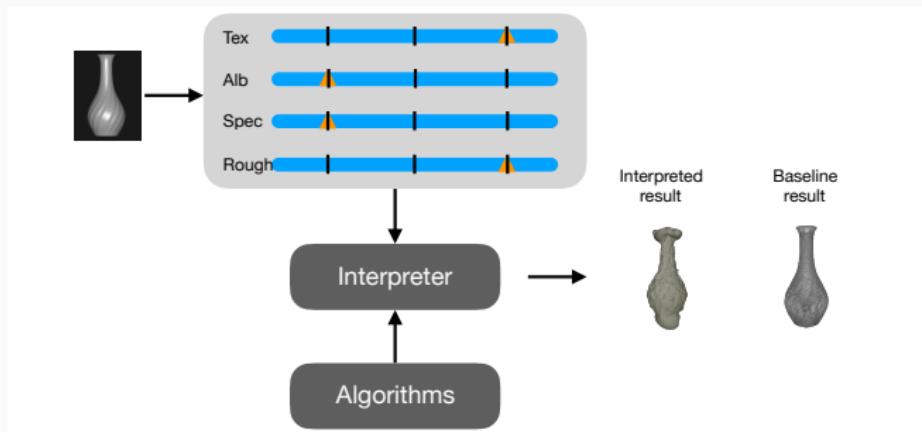
# Interpretation 2

## Summary

- Algorithm chosen by the interpreter given a less accurate description may or may not achieve a poor result;
- It depends if the return algorithms has at least an overlapping with those returned if given accurate description.

## Interpretation 3: demonstrative result

We use object 'vase' as an example:



## Summary

- accuracy: surface rough, inaccurate
- completeness:

## Interpretation 3: more results

More demonstrative results:

| Object         | Bust  | Vase0  | Barrel  | Vase1   |
|----------------|---|--|---|---|
| Incorrect Desc | 08020802<br>• BL<br>           | 08020208<br>• PMVS<br>• EPS<br> | 02020802<br>• BL<br>                     | 02020208<br>• EPS<br>           |
| Correct Desc   | 02080208<br>• EPS<br>• GSL<br> | 02080802<br>• EPS<br>           | 08080208<br>• PMVS<br>• EPS<br>• GSL<br> | 08080802<br>• PMVS<br>• EPS<br> |

## Interpretation 3: more results

| Object         | Statue  | Cup   | Pot   | Vase   |
|----------------|---|---|---|--|
| Incorrect Desc | 08020802<br>• BL  | 08020208<br>• PMVS<br>• EPS   | 02020802<br>• BL  | 02020208<br>• EPS  |
| Correct Desc   |  |  |  |  |

# Interpretation 3

## Summary

- Algorithm chosen by the interpreter given inaccurate description have a higher probability of achieving a poorer result;

## Conclusions

---

# Conclusions

- the proposed description is able to give correct reconstruction for non-concave objects
- To deal with more complicated objects, we need more complicated properties, or ways to describe the objects, but the challenge is the easy mathematical representation might not be available.
- Using the simple descriptive language and proof-of-concept interpreter, we demonstrate the possibility of using descriptive properties to hide algorithmic details.

Take-away

message

**Computer vision should focus on more than just algorithms, but easier accessibility.**