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

Kai Wu

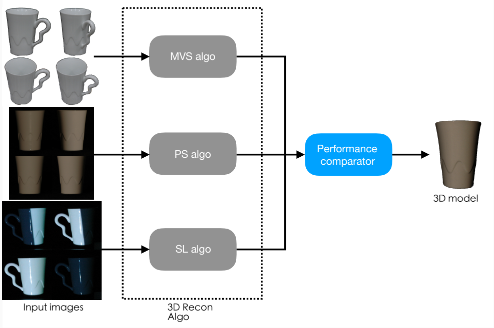
November 12, 2017

University of British Columbia
kaywu@ece.ubc.ca

Table of contents

Motivation

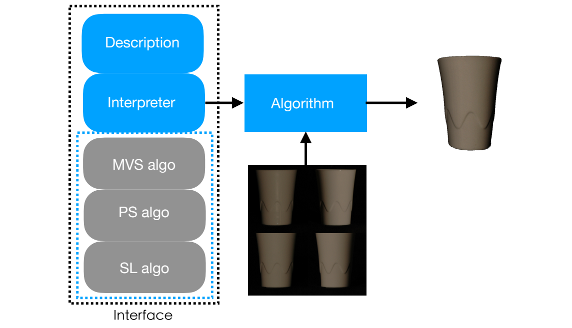
Motivation: traditional 3D reconstruction



Challenges

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

Motivation: interface to 3D reconstruction



Strengths

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

Contribution

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 interpreteded so that an appropriate algorithm is chosen to achieve a successful reconstruction result.

Contribution (cont'd)

This contribution is significant because:

- Few algorithms can work for a diverse categories of objects. The interface, to some extent, can cover a wider range of object categories by incorporating multiple algorithms.
- An description of object problem condition is provided to hide the algorithmic details, thus understanding of the algorithm, or conditions of applying algorithms are not a prerequisite.

Related Work

Related Work: softwares

Some noteable 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

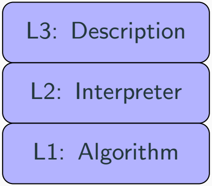


Figure 1: 3-layer interface to 3D reconstruction.

Description

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

Interpreter: translate description to an appropriate algorithm.

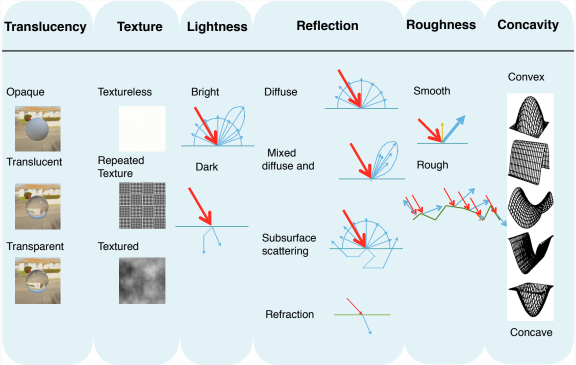
Mapping: discover the relation between problem space and algorithm.

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 works 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
	Textureless (Tl)	Textured (T)	Dark (D)	Bright (B)	Diffuse (D)	Mixed (M)	Smooth (S)	Rough (R)	
1	Yes			Yes	Yes			Yes	Tl-B-D-R
2	Yes			Yes		Yes	Yes		Tl-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	*Texture randomness*
Lightness	*Diffuse albedo*
Specularity	*Specular reflectance*
Roughness	*SD of facet slopes*

Table 1: Representations of the 3D reconstruction problem.

Description: expression

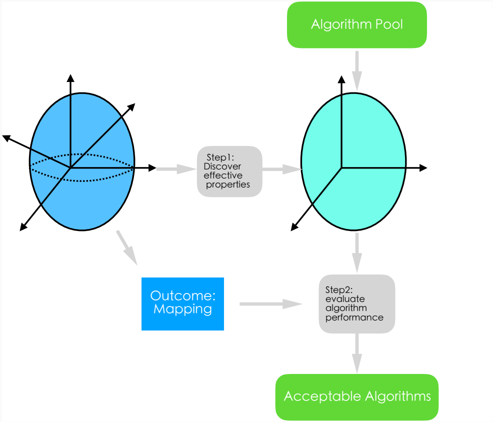
We use three discrete scales to parameterize these properties: *low* (0.2), *medium* (0.5), and *high* (0.8).

Prob cond	Texture	Albedo	Specular	Rough	**Label**
1	low/med	high	low/med	high	Tl-B-D-R
2	low/med	high	high	low/med	Tl-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

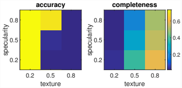
- Patch-based Multi-View Stereo (PMVS): propagate-refinement-filtering;
- Example-based Photometric Stereo (EPS): arbitrary BRDF is a linear combination of basis BRDFs;
- Gray-coded Structured Light (GSL): encode spatial informally temporally.

baseline methods

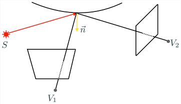
- Volumetric Visual Hull: carve voxels projecting outside of silhouettes;
- Linear-least squares Photometric Stereo: $\mathbf{I} = \rho \mathbf{N} \cdot \mathbf{L}$

Mapping: dataset creation

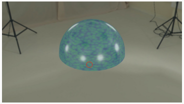
Mapping: notable findings 1



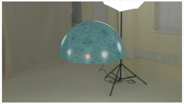
(a). Algo. performance



(b) Image formation



(c) V_1

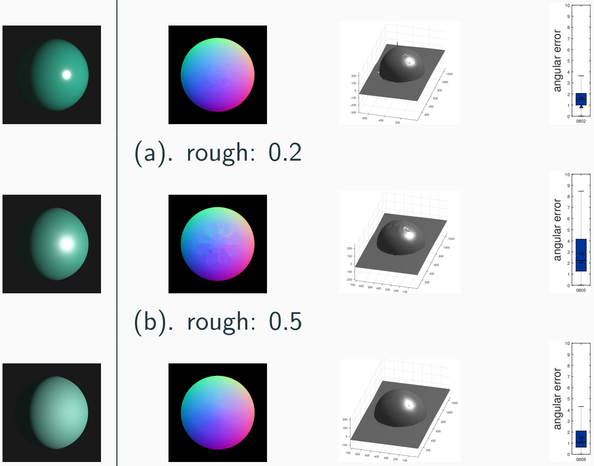


(d) V_2

Figure 2: (a) shows the algorithm performance w.r.t. texture and specularity. (b) shows the reflection of light off a specular surface. V_1 received the diffuse component while V_2 receives the specular component. (c), (d) shows the images observed from these two views. The specular area (red circle) observed in V_2 is visible in V_1.

Mapping: notable findings 2

Image	Normal map	Height map	Angular error



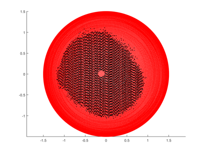
(a). rough: 0.2

(b). rough: 0.5

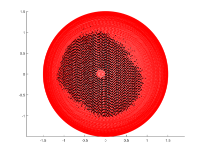
(c). rough: 0.8

Figure 3: The effect of roughness on PS. Albedo is set as 0.8, and specular is

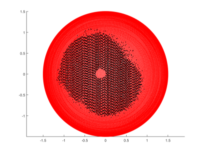
Mapping: notable findings 3



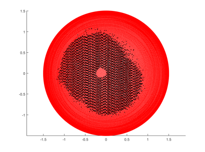
(a) specular: 0.2



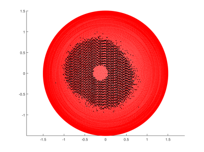
(b) specular: 0.5



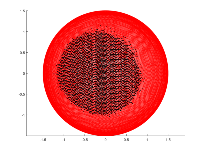
(c) specular: 0.8



(d) roughness: 0.2



(e) roughness: 0.5



(f) roughness: 0.8

Figure 4: (a)-(c): the roughness is set as 0.2, and specular has a negative effect on completeness; (d)-(e): the specular is set as 0.8, roughness has a positive effect on completeness.

Mapping: discussion

- PMVS can work on specular surfaces;
- EPS and GSL fails on highly specular areas, and a blurred specular area causes worse results.

Evaluation of interface

Interpretation: evaluation methodology

Evaluation question

1. Can the proof of concept interpreter return one of the best possible algorithms that achieves a successful reconstruction given the correct description?
2. Can a less accurate description give a less successful reconstruction result?
3. Can an inaccurate description give a poor reconstruction result?

Interpretation: evaluation methodology (cont'd)

Criteria

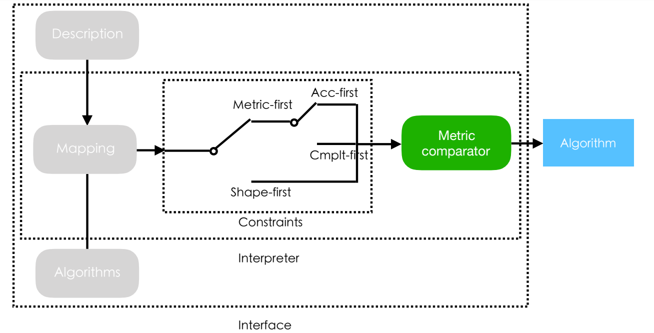
Visual comparison to results of baseline method.

Roadmap

- proof of concept interpreter;
- dataset creation;
- results of interpreter.

Interpretation: interpreter

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



Interpretation: dataset creation

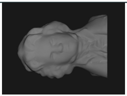
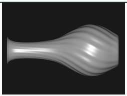
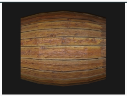
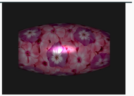
Calibration

method	calibration
MVS+VH	focal length from EXIF, extrinsics using SfM
PS	no radiometric calibration performed
SL	camera-projector calibration using local homography

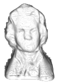
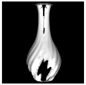
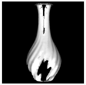
Creation

method	hardware	configuration
MVS+VH	camera	3 heights, 30° baseline angle
PS	camera, lamp, 2 ref objs	
SL	camera, projector	10° baseline angle

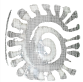
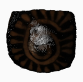
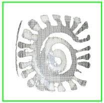
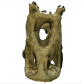
Interpretation: synthetic and real-world dataset

prob cond#	1	2	3	4
description	textureless diffuse bright	textureless mixed d/s bright	textured diffuse dark/bright	textured mixed d/s dark/bright
object				
				

Interpretation: evaluation of interpreter

Desc #	Bust	Vase1	Barrel	Vase0	Interp Algo.
1					GSL
2					EPS
3					GSL
4					PMVS

Interpretation: evaluation of interpreter (cont'd)

Desc #	Statue	Cup	Pot	Vase	Interp Algo.
1					GSL
2					EPS
3					GSL
4					PMVS

Interpretation: discussion

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.