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

Kai Wu

University of British Columbia

kaywu@ece.ubc.ca

October 4, 2017

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

Contribution (cont'd)

This contribution is significant because:

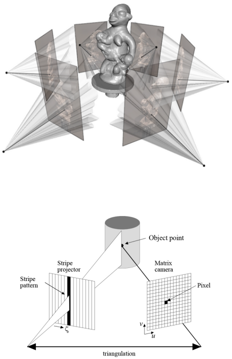
- No single algorithm can work for a diverse categories of objects. The interface, to some extent, can cover a wider range of object categories by incorporating multiple algorithms.
- An description is provides that hides the algorithmic details, thus understanding of the algorithm, or conditions to apply a specific algorithm is not a prerequisite.

Overview

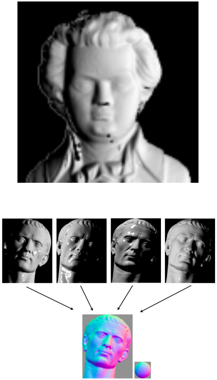
- **Taxonomy**: change the way of viewing algorithms, not from an algorithmic viewpoint, but from a problem-oriented (object-centered) viewpoint.
- **Description**: how to describe a sub-space in the $n-$dimensional problem space so that the conditions that an algorithm works reliably can be well defined;
- **Mapping**: discover the mapping between algorithms and the sub-spaces in the $n-$dimensional problem space.
- **Interpretation**: test the interpretability of the interface using synthetic and real-world datasets.

Related Work

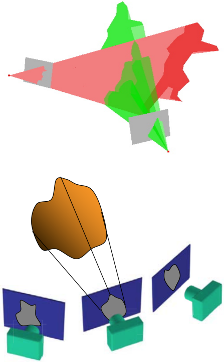
Previous algorithm taxonomy based on visual/geometric cues.



(a). Stereo



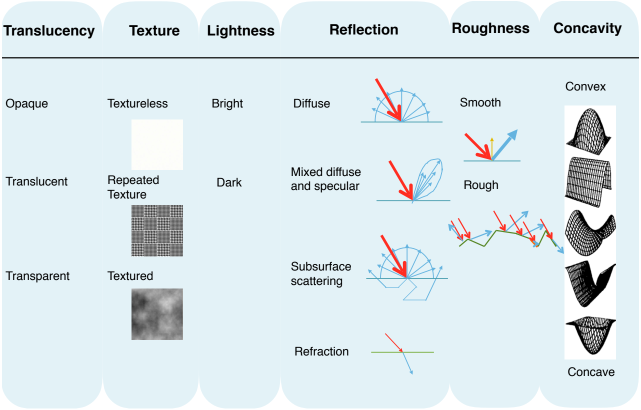
(b). Shading



(c) Silhouette

Taxonomy

- *algorithm-centered* taxonomy 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.



Taxonomy (cont'd)

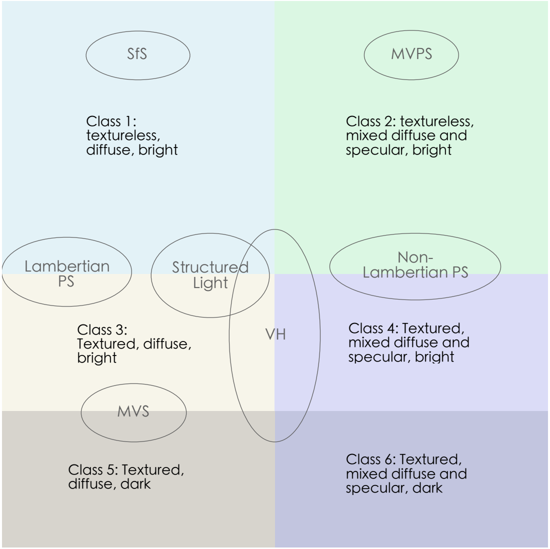
Analyze the working conditions of various classes of algorithms based on literature reports:

Algo.	Texture	Brightness	Reflectance	Roughness	Concavity
MVS	Textured	-	Mixed	-	-
SfS	-	Bright	Lambertian	-	Convex
L-PS	-	Bright	Lambertian	-	Convex
NL-PS	-	Bright	Mixed	-	Convex
SL	-	Bright	Diffuse	-	Convex
VH	-	-	-	-	-

Table: Problem conditions of six categories of algorithms based on literature reports. "L" stands for Lambertian, and "NL" for Non-Lambertian.

Taxonomy (cont'd)

The *object-centered* taxonomy of the six categories of algorithms.



Description: model and representations

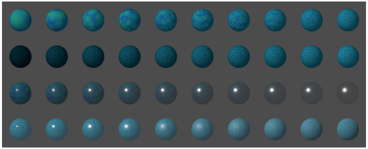
Model	**Representation**	**Visualization**
Texture	*Texture randomness*	
Lightness	*Albedo*	
Specularity	*Specular/diffuse ratio*	
Roughness	*SD of facet slopes*	
Concavity	*Surface curvature*	

Table: A Model and corresponding 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).

Object	Texture	Albedo	Specular	Rough	**Label**
Class 1	low/med	high	low/med	high	Tl-B-D-R
Class 2	low/med	high	high	low/med	Tl-B-M-S
Class 3	high	high	low/med	high	T-B-D-R
Class 4	high	high	high	low/med	T-B-M-S
Class 5	high	low/med	low/med	high	T-D-D-R
Class 6	high	low/med	high	low/med	T-D-M-S

Table: Expression of the reconstruction problem for the object classss.

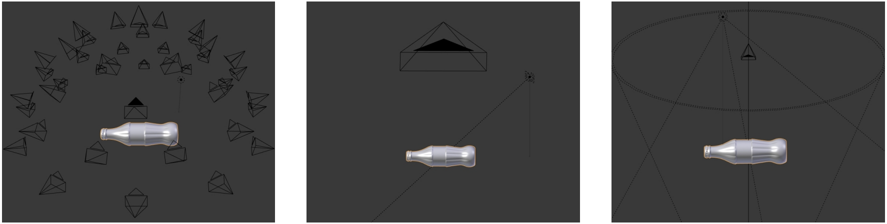
Mapping

Investigate the problem conditions under which the algorithms can reliably work. This strucure of this chapter is as follows

- Establish the *effective problem domain* (EPD): cope with large variation in material and shape.
- Evaluate algorithmic performance within EPD
- Derive mapping from problem conditions to algorithms.

Mapping: setup

Use Blender's physical-based render engine, Cycles to generate synthetic datasets.



(a). MVS (b). PS (c). SL

Figure: Synthetic setups.

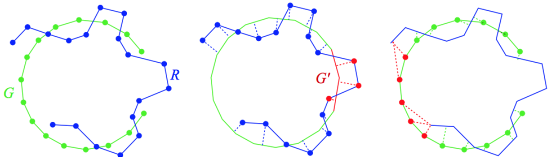
Mapping: algorithms and baseline

Technique	Texture	Albedeo	Specular	Roughness
PMVS: patch-based, seed points propagation MVS.	High	-	Low	-
EPS: example-based Photometric Stereo	-	High	Low	High
GSL: Gray-code Structured Light technique	-	High	Low	High
VH: volumetric Visual Hull.	-	-	-	-
LLS-PS: linear least squares Photometric Stereo.	-	High	Low	High

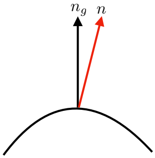
Table: Summary of the selected and baseline algorithms.

Mapping: quantitative measures and criteria

- accuracy: the distance d such that $X\%$ of the points on R are within distance d of G is considered as accuracy;
- completeness: the percentage of G that is reconstructed by R;
- angular error: angle between the estimated and ground truth normal, *i.e.*, $arccos(n_g^T n)$.



Accuracy and completeness



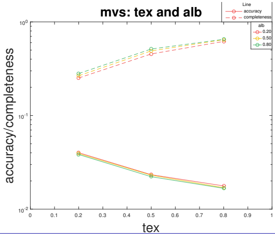
angular error

Mapping: EPD of PMVS

Table: Problem conditions for establishing the *effective problem domain* of PMVS.

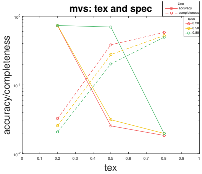
Cond	Texture	Albedo	Specular	Roughness
(a)	[0.2, 0.8]	[0.2, 0.8]	0.0	0.0
(b)	[0.2, 0.8]	0.8	[0.2, 0.8]	0.0
(c)	[0.2, 0.8]	0.8	0.0	[0.2, 0.8]
(d)	0.8	[0.2, 0.8]	[0.2, 0.8]	0.0
(e)	0.8	[0.2, 0.8]	0.0	[0.2, 0.8]
(f)	0.8	0.8	[0.2, 0.8]	[0.2, 0.8]

Cond. (a). Texture and Albedo

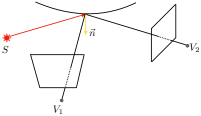


Mapping: EPD of PMVS (cont'd)

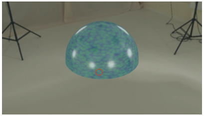
Cond. (b). Texture and Specularity



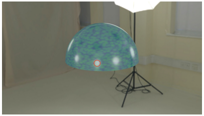
(a). Algo. performance



(b) Image formation



(c) V_1

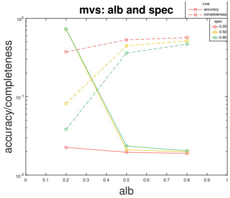


(d) V_2

Figure: (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: EPD of PMVS (cont'd)

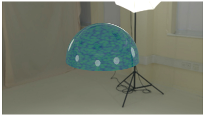
Cond. (d). Albedo and Specularity



(a) spec: 0.2

(b) spec: 0.8

(c) alb: 0.2



(d) alb: 0.8

Figure: The law of conservation of energy demands that as the specular reflection increases, the diffuse reflection decreases.

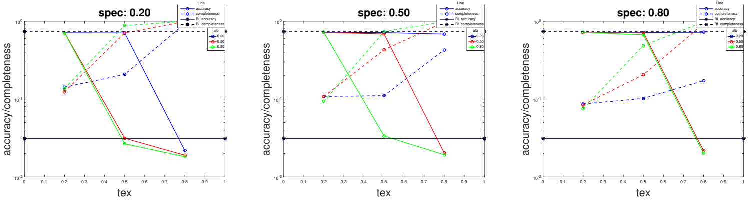
Mapping: *effective properties* of PMVS

Metric	Texture	Albedo	Specular	Roughness
Accuracy	✓	✓	✓	✗
Completeness	✓	✓	✓	✗

Table: The *effective problem domain* of PMVS in terms of accuracy and completeness.

Mapping: mapping construction

- Accuracy and completeness improves consistently as the *texture* level increases.
- Accuracy and completeness results deteriorate consistently as *specularity* increases, and this negative impact is most significant when texture level is medium or albedo value is low.
- The effect of *albedo* on a surface with low texture is negligible. However, albedo has a noticeably more significant positive impact on completeness as the texture of a specular surface increases.



Mapping: mapping construction (cont'd)

Metric	Texture	Albedo	Specular	Roughness
Accuracy	0.5	0.5	0.2	-
&Completeness	0.5	0.8	0.2	-
	0.5	0.8	0.5	-
	0.8	0.2	0.2	-
	0.8	0.5	0.2	-
	0.8	0.8	0.2	-
	0.8	0.5	0.5	-
	0.8	0.8	0.5	-
	0.8	0.5	0.8	-
	0.8	0.8	0.8	-

Table: The working problem conditions of PMVS in terms of the two metrics *accuracy* and *completeness*.

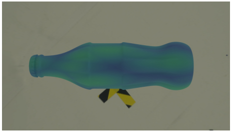
Interpretation

Key Evaluation Questions:

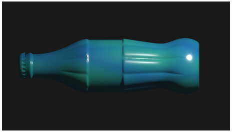
- Is the mapping robust to changes of shape?
- Can the proof-of-concept interpreter return a satisfactory reconstruction given the correct description?

Interpretation: evaluation of mapping

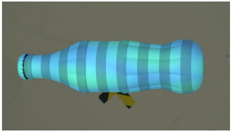
- Shape variation: too vast and complicated to model
- Instead focus on one geometric property: surface concavity.



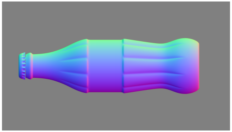
MVS



PS

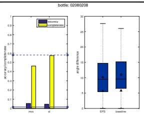
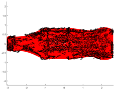
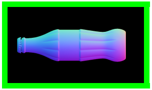
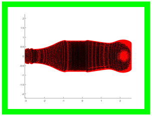
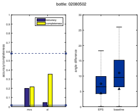
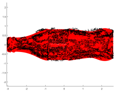
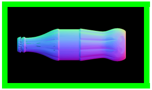
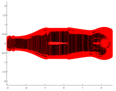


SL

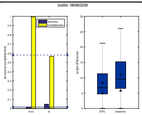
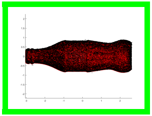
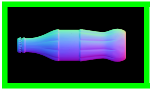
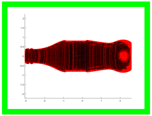
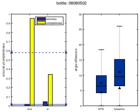
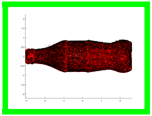
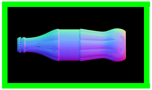
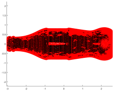


Normal groundtruth

Interpretation: evaluation of mapping

Mapping	Quantitative results		Qualitative results	
EPS, GSL				
	(a). tex(0.2), alb(0.8), spec(0.2), rough(0.8)			
EPS				
	(b). tex(0.2), alb(0.8), spec(0.5), rough(0.2)			
		MVS	PS	SL

Interpretation: evaluation of mapping (cont'd)

Mapping	Quantitative results		Qualitative results	
PMVS, EPS, GSL				
	(c). tex(0.8), alb(0.8), spec(0.2), rough(0.8)			
PMVS, EPS				
	(d). tex(0.8), alb(0.8), spec(0.5), rough(0.2)			
		MVS	PS	SL

Proof-of-concept interpreter

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

- depth-first:
- quality-first:

The default if depth-first, therefore, if multiple algorithm are selected, the typical order is GSL $>$ PMVS $>$ EPS.

Interpretation: evaluation of interpreter

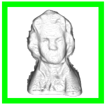
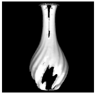
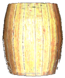
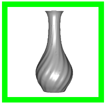
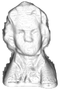
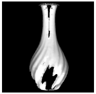
Desc #	Bust	Vase1	Barrel	Vase0	Selected Algo.
1					GSL
2					EPS
3					GSL
4					PMVS

Figure: The evaluation of interpreter using synthetic objects. The first column presents the description provided to the interpreter. Description i matches with condition i. The last column is the algorithm selected by the interpreter. The object of which condition matches the description is labeled in green rectangle.

Interpretation: real-world objects

class #	1	2	3&4	5&6
description	textureless diffuse bright	textureless mixed d/s bright	textured diffuse dark/bright	textured mixed d/s dark/bright
object				

Figure: The rerepsentatives of the six classes of objects used for evaluation.

Interpretation: evaluation of interpreter (cont'd)

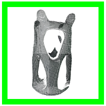
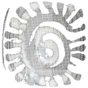
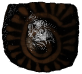
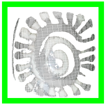
Desc #	Statue	Cup	Pot	Vase	Selected Algo.
1					GSL
2					EPS
3					GSL
4					PMVS

Figure: The evaluation of interpreter using real-world objects. The first column presents the description provided to the interpreter. Description *i* matches with condition *i*. The last column is the algorithm selected by the interpreter. The object of which the condition matches the description is labeled in green

Conclusions

- Using the simple descriptive language and proof-of-concept interpreter, we demonstrate the possibility of using descriptive properties to hide algorithmic details.

Reconstructed models

Questions?