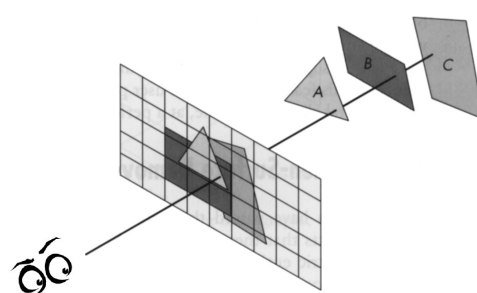



1

Rimozione delle superfici nascoste (cioè occluse)

✓ Gli oggetti più vicini dal POV devono coprire (“occludere”) quelli più lontani



Usando raytracing, questo effetto è ottenuto in modo ovvio!
Usando rasterization?



2

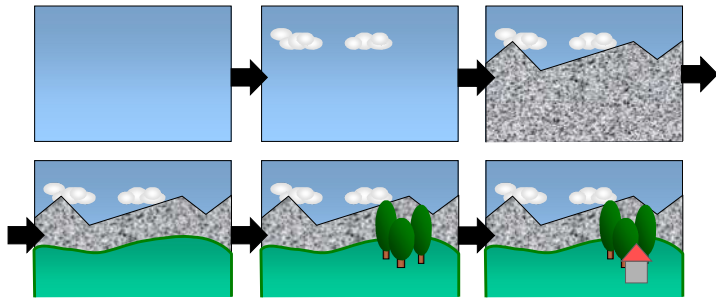
Rimozione delle superfici nascoste ("hidden surface removal")

- ✓ Negli approcci basati su ray-tracing, questo è banale da ottenere:
 - ⇒ basta prendere, per ogni raggio primario, la primitiva intersecata avente la distanza k minore
- ✓ Negli approcci basati sulla rasterizzazione, non è così banale
 - ⇒ Se i pixel prodotti dalla rasterizzazione di una primitiva **sovrascrivono** sullo screen buffer quelli generati (alla stessa posizione) dalle primitive precedenti...
 - ⇒ ...allora il valore finale sullo screen buffer sarà semplicemente il valore dei pixel generati dall'ultima primitiva rasterizzata
 - ⇒ ...e perché questo risultato sia quello corretto, è necessario rasterizzare per ultime le primitive più vicine

3

Rimozioni delle superfici nascoste tramite ordinamento

- ✓ Soluzione basata su ordinamento
 - ⇒ le primitive rasterizzate **sovrascrivono** nel frame buffer quelle rasterizzate in precedenza
 - ⇒ quindi, basta rasterizzare le primitive nell'*ordine giusto* ordine detto "back-to-front"
 - ⇒ approccio detto "**algoritmo del pittore**"



5

Algoritmo del pittore (o depth sorting)

✓ Data una scena (composta da primitive)

1. ordinare le primitive back-to-front

- dalle più lontane dalla camera, alle più vicine

⇒ quindi, in ordine di coordinata Z...

- *crescente*, se in **spazio vista**
dato che la Z del nostro spazio vista decresce col crescere della distanza dalla camera
- *decrescente*, in **spazio clip** o **spazio schermo**
dato che, la Z cresce col decrescere della distanza dalla camera
(in spazio schermo: è detta «depth» e va da 0 a 1;
in spazio clip: va da -1 a +1)

2. rasterizzarle in successione

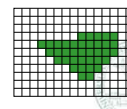
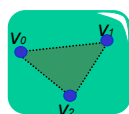
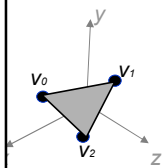
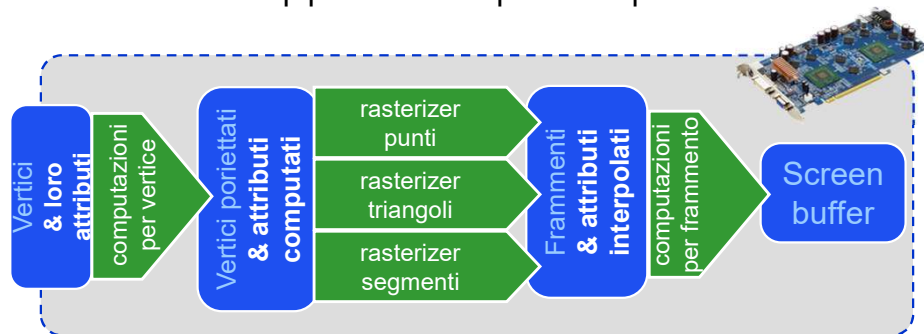
- Permettendo a tutti frammenti generati di sovrascrivere i pixel già presenti sul buffer (prodotti dai frammenti generati in precedenza)



6

Algoritmo del pittore (depth sorting): non è adatto a soluzioni hardware

La coordinata z in spazio vista non è nota a priori.
Nessuna fase del pipeline HW si presta a questo task.



7

Algoritmo del pittore (depth sorting)

- ✓ In questo esempio, la primitiva A può essere disegnata tranquillamente per prima
- ✓ Ma va prima C o D?

Depth (z in screen space)

Primitive

A

B

C

D

E

Z_{max}

Z_{min}

8

Algoritmo del pittore (depth sorting)

- ✓ Esiste sempre un ordine corretto? NO

- ⇒ Controesempio: intersezioni

- ⇒ Controesempio: cicli di occulsione

A

B

C

D

E

Z_{max}

Z_{min}

9

Rimozione delle superfici nascoste tramite ordinamento su Z: sommario degli ostacoli

- ✓ Limiti delle soluzioni basate su ordinamento:
 - ⇒ Problema 1: ordinamento costa $O(n \log n)$ (è "pseudolineare")
 - con n numero di primitive
 - n può molto grande. Es: se $n = \text{milioni}$ → $\log(n)$ è un fattore 30
In CG, qualsiasi cosa peggiore di lineare è mal tollerato
 - ⇒ Problema 2: quando effettuare l'ordinamento?
 - le coordinate in spazio vista (compreso la Z) sono note solo dopo la trasformazione
 - ⇒ Problema 3: una primitiva non ha un'unica Z
 - ogni vertice che la compone ha una Z diversa
 - quale usare? (media, min, max...). Scelte diverse, ordini diversi.
 - a volte, non esiste alcun ordinamento che dà i risultati corretti
 - ⇒ Problema 4: complica il rendering
 - prescrivendo un certo ordine di rendering delle primitive



10

Algoritmo del pittore (depth sorting)


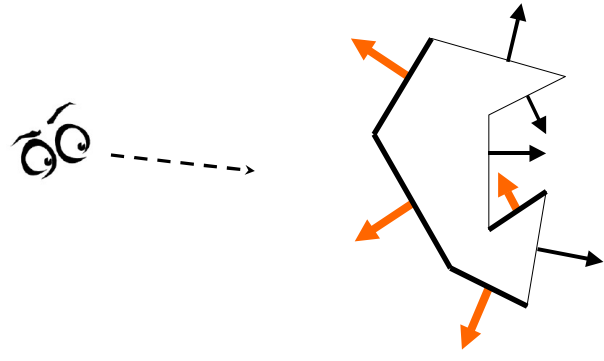
- ✓ Gli ostacoli sono superabili, ma l'algoritmo del pittore in pratica è usato raramente
 - ⇒ solo nelle occasioni in cui sia facile ordinare preventivamente le primitive back-to-front in fase di preprocessing
 - ⇒ esempio: mesh ottenuta poligonalizzando un campo di altezza
- ✓ Vediamo invece alternative per rimuovere le superfici nascoste che sono «*order independent*»
 - ⇒ Il rendering produce lo stesso risultato, *indipendentemente* dall'ordine delle primitive



11

Caso particolare per mesh chiuse e ben orientate. Backface Culling

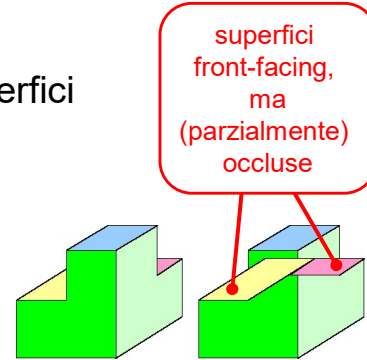
- ✓ E' un caso particolare di rimozione delle superfici nascoste, che abbiamo già visto
- ✓ Idea: se la superficie **chiusa** e **opaca**...
(e la camera è all'esterno)
allora, non ne vedrò mai l'interno!




12

Il back-face culling non basta a rimuovere *tutte* le superfici nascoste

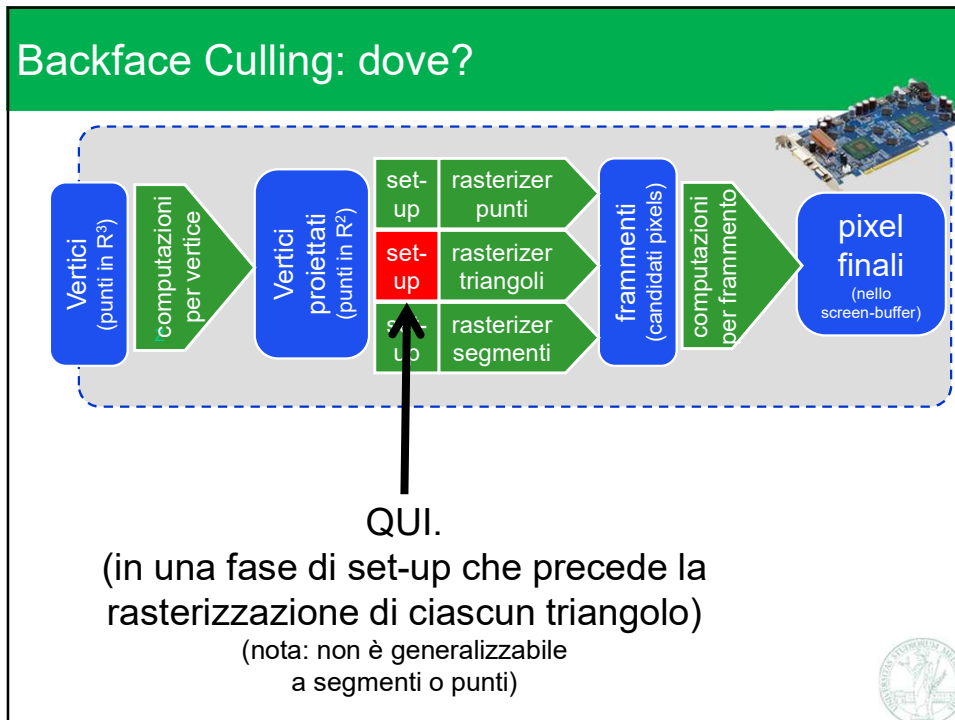
- ✓ Contro-esempio:
⇒ succede solo con le superfici concave, cioè non convesse



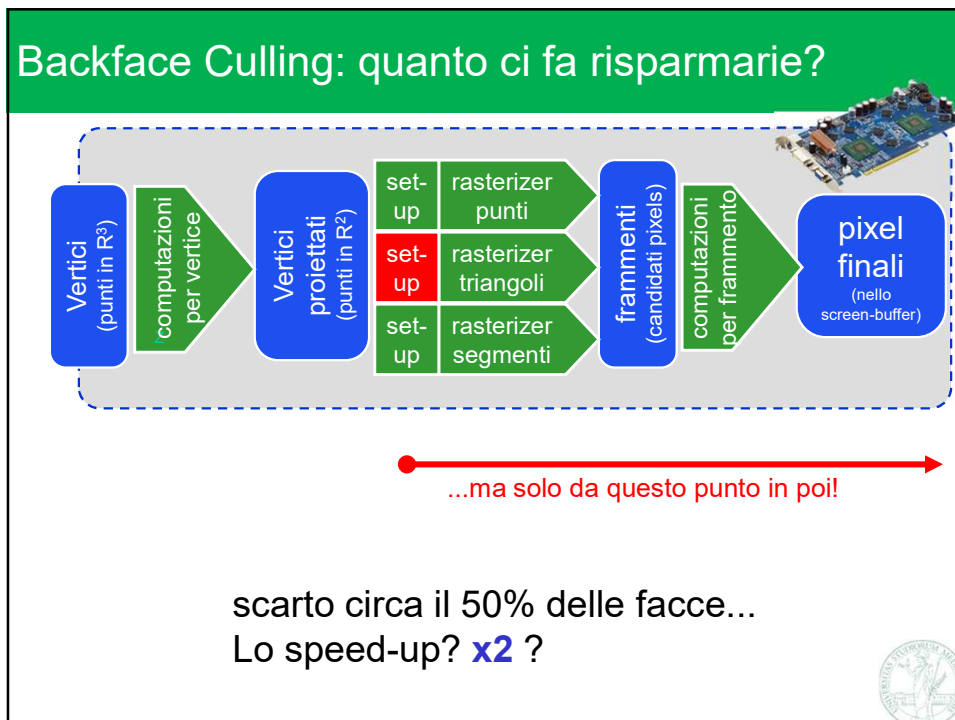
- ✓ Tuttavia, è utile come ottimizzazione:
⇒ mi consente di non processare alcune primitive, perché so che verranno comunque occluse



13



14



15

Quando usare il backface culling (sommario)

- ✓ *Se valgono le ipotesi dette*, allora il back-face culling non cambia le immagini generate
 - ⇒ In caso contrario, genera artefatti (scompare il «retro» delle superfici)
- ✓ In questo caso, è conveniente perché...
 - ⇒ migliora le prestazioni delle applicazioni fill-limited, riducendo di circa il 50% il numero di frammenti prodotti e da processare
 - Non impatta però le applicazioni geometry-limited, dato che lo scarto della primitiva avviene a valle del processing per vertice
 - ⇒ contribuisce all'hidden-surface removal, dato rimuove solo facce occluse
 - Non è però sufficiente, nel caso di mesh concave, perché non rimuove *tutte* le facce auto-occluse (vedi dopo)
- ✓ E' un test eseguito in automatico dall'HW della GPU
 - ⇒ Quindi, programmando l'API a basso livello (come WebGL o OpenGL), il programmatore si limita ad abilitarlo oppure disabilitarlo



18

Esempio di abilitazione del backface culling in librerie ad alto livello: in three.js (JavaScript)

- ✓ Abilito o disabilito il back-face culling settando un parametro del “materiale” associato alle mesh
 - ⇒ Back-Face Culling abilitato, scarta le face back-facing:

```
materiale.side = THREE.Front;
```

il default
 - ⇒ “Front-Face Culling” abilitato, scarta le face front-facing:

```
materiale.side = THREE.Back;
```
 - ⇒ Back-Face Culling disabilitato:


```
materiale.side = THREE.DoubleSide;
```



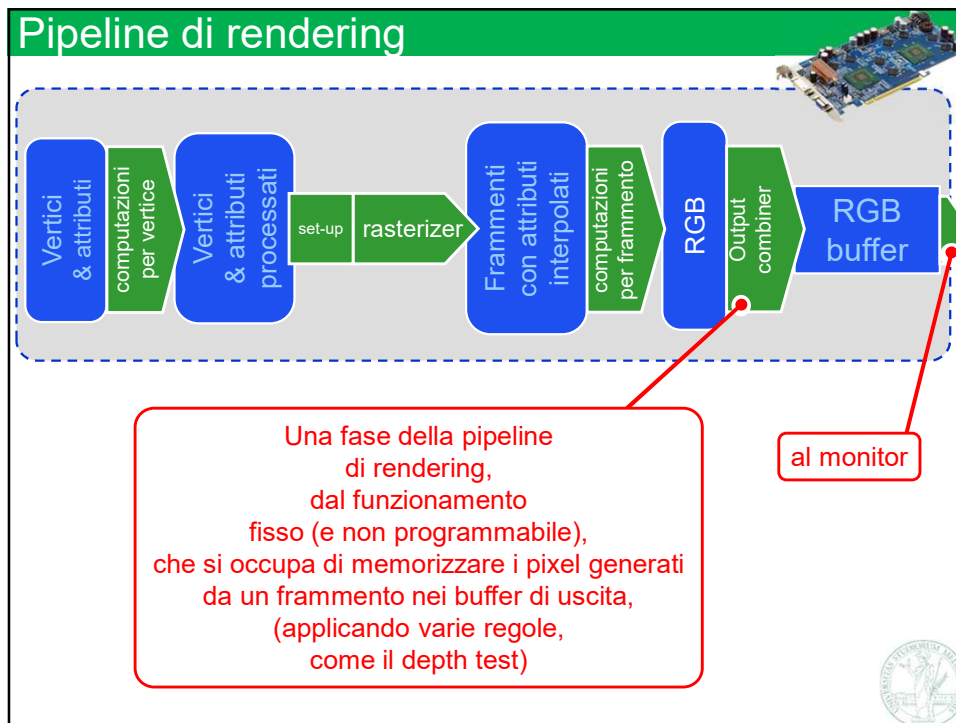
20

Rimozione delle superfici nascoste: attraverso il Depth-buffer

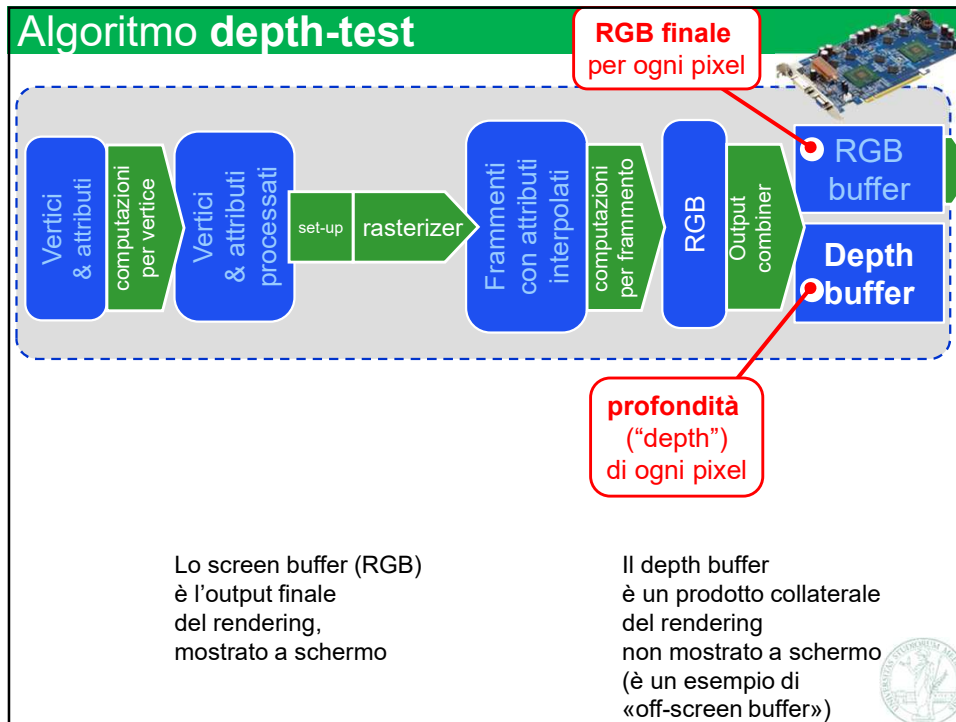
- ✓ Idea: eseguire un test a livello di frammento
 - ⇒ Di tutti i frammenti che insistono su un pixel, voglio tenere solo quello di profondità (depth) minore
 - cioè quello più vicino all'osservatore
 - ⇒ Serve una struttura per memorizzare la profondità corrente di ogni pixel...
- ✓ «Depth-buffer»: (a volte: «Z-buffer»)
 - ⇒ un buffer 2D (una griglia, una matrice di valori)
 - ⇒ stessa risoluzione dello screen buffer (per ogni pixel, ho un valore di depth)
 - ⇒ per ogni posizione [x , y] memorizza il depth del frammento più vicino all'osservatore (di depth minore) che sia stato scritto a [x , y]



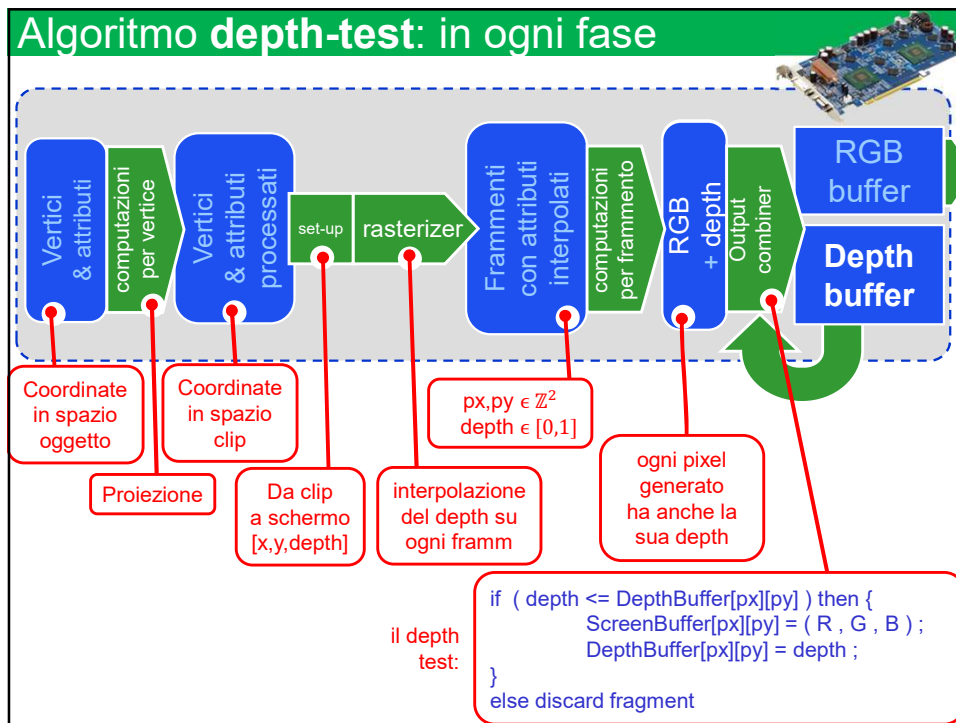
21



22



23



24

Valore di depth (profondità) di un frammento

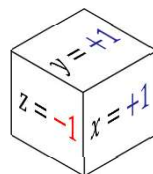
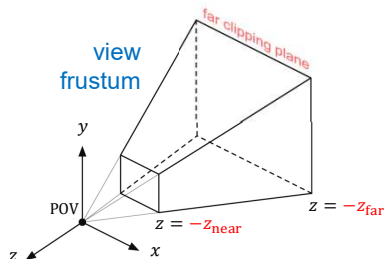
- ✓ E' un valore da 0 a 1
 - ⇒ 0: pixel più vicino possibile all'osservatore
 - ⇒ 1: pixel più lontano possibile dall'osservatore
 - ⇒ i frammenti con depth non inclusa fra 0 e 1 sono scartati automaticamente dal rasterizzatore (il pixel non è nel view frustum, è scarato dal far o near clipping plane)
- ✓ E' un sinonimo della la coordinata Z dello «spazio schermo»
 - ⇒ Lo spazio in cui un pixel ha coordinate intere in x e y che vanno da 0 a risoluzione immagine (in larghezza e altezza risp.)
 - ⇒ Vedi lezione sulla proiezione e rasterizzazione
- ✓ E' calcolato dalla Z in spazio clip: $depth = (z_{CLIP} + 1)/2$
- ✓ E' interpolato, dentro ogni primitiva, per ottenere il valore di *depth* di ogni frammento
 - ⇒ Attraverso coordinate baricentriche
 - ⇒ Come qualsiasi altro valore che sia definito sui vertici



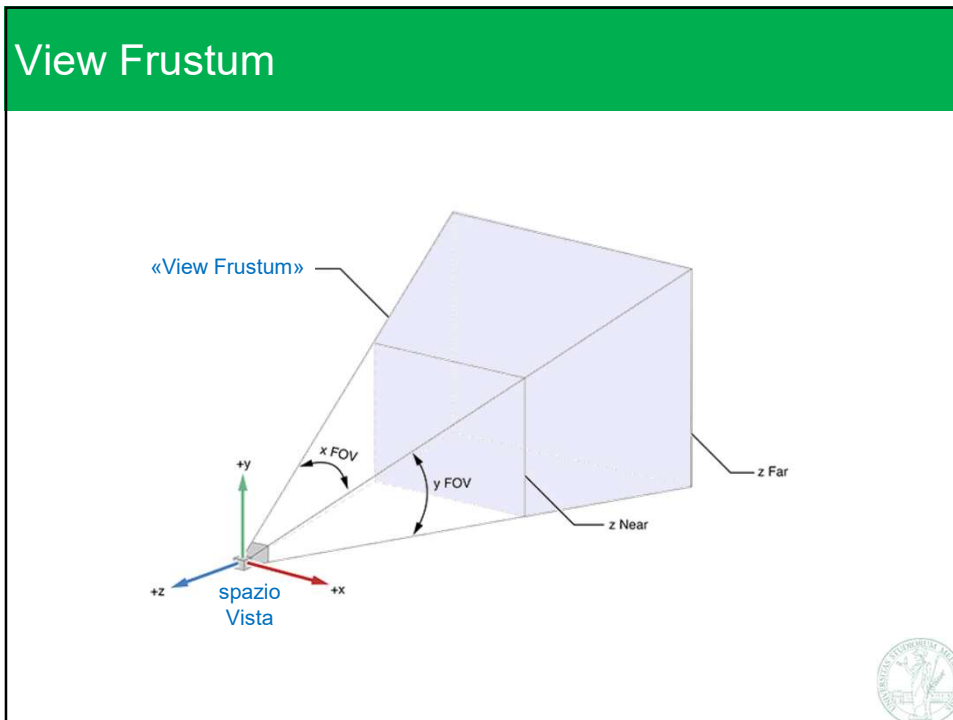
26

Limiti dello spazio visibile (cioè lo spazio inquadrato) in vari spazi

	Spazio Vista	Spazio Clip	Spazio Schermo	
x	da: «left clipping plane»	-1	0	} pixel coords
	a: «right clipping plane»	+1	res_x	
y	da: «bottom clipping plane»	-1	0	
	a: «top clipping plane»	+1	res_y	
z	da: $-z_{near}$ «near clipping plane»	-1	0	} pixel depth
	a: $-z_{far}$ «far clipping plane»	+1	1	



27





28

Algoritmo dello depth-test: esempio


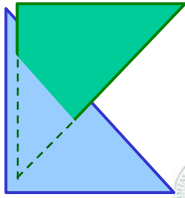
<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td></tr> <tr><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td></tr> <tr><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td></tr> <tr><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td></tr> <tr><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td></tr> <tr><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td></tr> <tr><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td></tr> <tr><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td></tr> </table>	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	+	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td></td></tr> <tr><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td></td><td></td></tr> <tr><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td></td><td></td><td></td></tr> <tr><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td></td><td></td><td></td><td></td></tr> <tr><td>.5</td><td>.5</td><td>.5</td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>.5</td><td>.5</td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>.5</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>.5</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>	.5	.5	.5	.5	.5	.5	.5		.5	.5	.5	.5	.5	.5			.5	.5	.5	.5	.5				.5	.5	.5	.5					.5	.5	.5						.5	.5							.5								.5								=	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>1.0</td></tr> <tr><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>1.0</td><td>1.0</td></tr> <tr><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>1.0</td><td>1.0</td><td>1.0</td></tr> <tr><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td></tr> <tr><td>.5</td><td>.5</td><td>.5</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td></tr> <tr><td>.5</td><td>.5</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td></tr> <tr><td>.5</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td></tr> <tr><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td></tr> </table>	.5	.5	.5	.5	.5	.5	.5	1.0	.5	.5	.5	.5	.5	.5	1.0	1.0	.5	.5	.5	.5	.5	1.0	1.0	1.0	.5	.5	.5	.5	1.0	1.0	1.0	1.0	.5	.5	.5	1.0	1.0	1.0	1.0	1.0	.5	.5	1.0	1.0	1.0	1.0	1.0	1.0	.5	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0																																																																																																																																																																																													
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0																																																																																																																																																																																													
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0																																																																																																																																																																																													
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0																																																																																																																																																																																													
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0																																																																																																																																																																																													
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0																																																																																																																																																																																													
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0																																																																																																																																																																																													
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0																																																																																																																																																																																													
.5	.5	.5	.5	.5	.5	.5																																																																																																																																																																																														
.5	.5	.5	.5	.5	.5																																																																																																																																																																																															
.5	.5	.5	.5	.5																																																																																																																																																																																																
.5	.5	.5	.5																																																																																																																																																																																																	
.5	.5	.5																																																																																																																																																																																																		
.5	.5																																																																																																																																																																																																			
.5																																																																																																																																																																																																				
.5																																																																																																																																																																																																				
.5	.5	.5	.5	.5	.5	.5	1.0																																																																																																																																																																																													
.5	.5	.5	.5	.5	.5	1.0	1.0																																																																																																																																																																																													
.5	.5	.5	.5	.5	1.0	1.0	1.0																																																																																																																																																																																													
.5	.5	.5	.5	1.0	1.0	1.0	1.0																																																																																																																																																																																													
.5	.5	.5	1.0	1.0	1.0	1.0	1.0																																																																																																																																																																																													
.5	.5	1.0	1.0	1.0	1.0	1.0	1.0																																																																																																																																																																																													
.5	1.0	1.0	1.0	1.0	1.0	1.0	1.0																																																																																																																																																																																													
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0																																																																																																																																																																																													
<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>1.0</td></tr> <tr><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>1.0</td><td>1.0</td></tr> <tr><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>1.0</td><td>1.0</td><td>1.0</td></tr> <tr><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td></tr> <tr><td>.5</td><td>.5</td><td>.5</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td></tr> <tr><td>.5</td><td>.5</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td></tr> <tr><td>.5</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td></tr> <tr><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td></tr> </table>	.5	.5	.5	.5	.5	.5	.5	1.0	.5	.5	.5	.5	.5	.5	1.0	1.0	.5	.5	.5	.5	.5	1.0	1.0	1.0	.5	.5	.5	.5	1.0	1.0	1.0	1.0	.5	.5	.5	1.0	1.0	1.0	1.0	1.0	.5	.5	1.0	1.0	1.0	1.0	1.0	1.0	.5	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	+	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>.7</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>.6</td><td>.7</td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>.5</td><td>.6</td><td>.7</td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>.4</td><td>.5</td><td>.6</td><td>.7</td><td></td><td></td><td></td><td></td></tr> <tr><td>.3</td><td>.4</td><td>.5</td><td>.6</td><td>.7</td><td></td><td></td><td></td></tr> <tr><td>.2</td><td>.3</td><td>.4</td><td>.5</td><td>.6</td><td>.7</td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>	.7								.6	.7							.5	.6	.7						.4	.5	.6	.7					.3	.4	.5	.6	.7				.2	.3	.4	.5	.6	.7																			=	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>1.0</td></tr> <tr><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>1.0</td><td>1.0</td></tr> <tr><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>1.0</td><td>1.0</td><td>1.0</td></tr> <tr><td>.5</td><td>.5</td><td>.5</td><td>.5</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td></tr> <tr><td>.4</td><td>.5</td><td>.5</td><td>.7</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td></tr> <tr><td>.3</td><td>.4</td><td>.5</td><td>.6</td><td>.7</td><td>1.0</td><td>1.0</td><td>1.0</td></tr> <tr><td>.2</td><td>.3</td><td>.4</td><td>.5</td><td>.6</td><td>.7</td><td>1.0</td><td>1.0</td></tr> <tr><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td></tr> </table>	.5	.5	.5	.5	.5	.5	.5	1.0	.5	.5	.5	.5	.5	.5	1.0	1.0	.5	.5	.5	.5	.5	1.0	1.0	1.0	.5	.5	.5	.5	1.0	1.0	1.0	1.0	.4	.5	.5	.7	1.0	1.0	1.0	1.0	.3	.4	.5	.6	.7	1.0	1.0	1.0	.2	.3	.4	.5	.6	.7	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
.5	.5	.5	.5	.5	.5	.5	1.0																																																																																																																																																																																													
.5	.5	.5	.5	.5	.5	1.0	1.0																																																																																																																																																																																													
.5	.5	.5	.5	.5	1.0	1.0	1.0																																																																																																																																																																																													
.5	.5	.5	.5	1.0	1.0	1.0	1.0																																																																																																																																																																																													
.5	.5	.5	1.0	1.0	1.0	1.0	1.0																																																																																																																																																																																													
.5	.5	1.0	1.0	1.0	1.0	1.0	1.0																																																																																																																																																																																													
.5	1.0	1.0	1.0	1.0	1.0	1.0	1.0																																																																																																																																																																																													
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0																																																																																																																																																																																													
.7																																																																																																																																																																																																				
.6	.7																																																																																																																																																																																																			
.5	.6	.7																																																																																																																																																																																																		
.4	.5	.6	.7																																																																																																																																																																																																	
.3	.4	.5	.6	.7																																																																																																																																																																																																
.2	.3	.4	.5	.6	.7																																																																																																																																																																																															
.5	.5	.5	.5	.5	.5	.5	1.0																																																																																																																																																																																													
.5	.5	.5	.5	.5	.5	1.0	1.0																																																																																																																																																																																													
.5	.5	.5	.5	.5	1.0	1.0	1.0																																																																																																																																																																																													
.5	.5	.5	.5	1.0	1.0	1.0	1.0																																																																																																																																																																																													
.4	.5	.5	.7	1.0	1.0	1.0	1.0																																																																																																																																																																																													
.3	.4	.5	.6	.7	1.0	1.0	1.0																																																																																																																																																																																													
.2	.3	.4	.5	.6	.7	1.0	1.0																																																																																																																																																																																													
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0																																																																																																																																																																																													

29

Algoritmo del depth-test: vantaggi

- ✓ il rendering diventa "order independent" 😊!!!
- ✓ Funziona su tutto 😊
⇒ anche su:

- ✓ Adatto all'implementazione parallela quindi HW 😊


.5	.5	.5	.5	.5	.5	.5	1.0
.5	.5	.5	.5	.5	.5	1.0	1.0
.5	.5	.5	.5	.5	1.0	1.0	1.0
.5	.5	.5	.5	1.0	1.0	1.0	1.0
.4	.5	.5	.7	1.0	1.0	1.0	1.0
.3	.4	.5	.6	.7	1.0	1.0	1.0
.2	.3	.4	.5	.6	.7	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0



30

Algoritmo del depth-test: costi e limiti

- ✓ Costa un po' di memoria (GPU)
⇒ per memorizzare lo il depth buffer
⇒ il buffer deve essere inizializzato (a profondità massima) prima di ogni rendering
- ✓ Problemi di *aliasing sulla z*
⇒ detto: "z-fighting"
⇒ quando la precisione dei valori di depth non è sufficiente
es., se si renderizzano due superfici parallele molto vicine
- ✓ I frammenti vengono scartati dal test tardi nel pipeline
⇒ quando tutta la computazione precedente è già stata inutilmente effettuata
- ✓ Assume superfici del tutto *opache*
⇒ problemi con le superfici *semitrasparenti*
- ✓ Il *depth buffer* è memoria condivisa in lettura e scrittura 😊
⇒ complicazione per chi implementa HW
⇒ efficienza ameno in parte impattata
(anche se i produttori HW mettono in atto molte ottimizzazioni)
⇒ per questo, il depth test è gestito da una apposita fase non programmabile
(a valle delle computazioni programmabili per frammento) detta output combiner



32

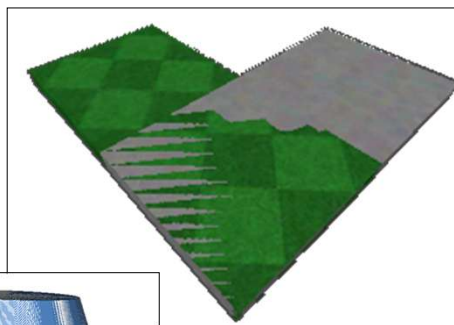
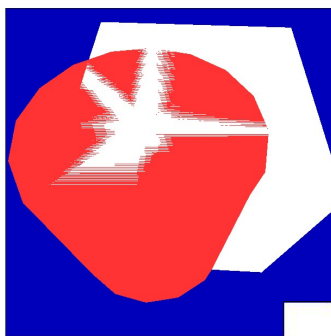
Depth test: problemi di precisione

- ✓ Il depth buffer memorizza valori scalari da 0 a 1 inclusi
- ✓ Possono essere rappresentati in virgola mobile, ma più spesso sono in virgola fissa, senza segno
- ✓ Es: se usassi 8 bit, allora, memorizzo l'intero $d \in \mathbb{Z}$ (da 0 a 255) per rappresentare il razionale $d / 255 \in \mathbb{Q}$
- ✓ La precisione necessaria al test impone di usare più di 8 bit: almeno 16
 - ⇒ Servono ben più di $2^8 = 256$ livelli di profondità distinti, per evitare z-fighting
- ✓ Il problema di z-fighting non viene mai completamente evitato, e si manifesta se vengono renderizzate superfici parallele sufficientemente vicine fra loro (oppure, peggio ancora, coincidenti in teoria)



33

Z-fighting: esempi



Due mesh che modellano trochi di cono: una azzurra e una, *leggermente* più piccola, bianca



34

Effetto collaterale del Depth test: ogni rendering produce anche una depth image

- ✓ Dopo ogni rendering,
ho prodotto non solo un'immagine a colori (RGB per pixel)
ma anche un depth-buffer (profondità per pixel)
 - ⇒ una depth image che rappresenta un «bassorilievo» della scena
 - ⇒ Si tratta quindi di una range scan «virtuale»



- ⇒ il color-buffer (RGB) viene mandato a schermo,
- ⇒ il depth-buffer è solo di uso interno e normalmente viene scartato
- ⇒ alcuni algoritmi di rendering lo sfruttano invece per gli scopi più vari



37

Depth test, vantaggi

- ✓ Il rendering è reso *order independent*:
 - ⇒ se disegno prima la primitiva davanti:
i frammenti di quella dietro verranno scartati
 - ⇒ se disegno prima la primitiva dietro:
i frammenti della primitiva davanti
li sovrascriveranno
 - ⇒ risultato finale: è lo stesso!
 - ⇒ L'efficienza può essere però diversa:
scartare è più efficiente di
scrivere-per-poi-sovrascrivere
 - Quindi il depth sorting è ancora utile, ma solo come ottimizzazione
 - L'ordine ottimale è front-to-back
 - L'opposto di quello usato nell'algoritmo del pittore!



39

Abilitare il depth test

- ✓ Il depth test è implementato in HW nella GPU
- ✓ Quindi, sia nelle API grafiche a basso livello che (a maggior ragione) nelle librerie ad alto livello, il programmatore si limita ad abilitarlo oppure disabilitarlo
 - ⇒ Nelle API a basso livello, è anche responsabilità del programmatore cancellare il depth buffer (cioè settarlo al valore massimo, 1.0) all'inizio di ogni rendering
- ✓ In three.js, questa scelta è effettuata settando un apposito flag nel materiale della mesh renderizzata
 - ⇒ `mioMateriale1.depthTest = true;`
 - ⇒ Di default, è abilitata
- ✓ In molti contesti, come video-gaming, il depth-test si assume, di norma, sempre abilitato



43

Considerazioni pratiche sulla scelta di zNear e zFar nella matrice di Proiezione

- ✓ a volte è utile scartare il troppo vicino (attraverso **zNear**)
 - ⇒ es, se telecamera è «dentro» un muro: non voglio vedere il muro
 - ⇒ telecamera dentro la testa di un personaggio: non voglio vedere la testa
 - ⇒ muro o testa coprirebbero l'intera scena!
- ✓ non è quasi mai utile scartare il troppo lontano (attraverso **zFar**)
 - ⇒ gli oggetti appaiono più piccoli con la distanza, ma mai fino a sparire
 - ⇒ perché ometterli del tutto?
 - ⇒ dunque, idealmente, $zFar = +\infty$ ma...
- ✓ entrambe i limiti alla z sono comunque necessari
 - ⇒ Se zFar grande, o zNear piccolo → grande intervallo di Z vista mappato nel intervallo depth da 0 a 1 → precisione numerica si riduce → z Fighting ☹
- ✓ nota: **$zFar > zNear > 0$** sempre (per definizione)
- ✓ esempio di scelta ragionevole:
«inquadro da 20 cm a 50 metri» → $zNear = 0.2, zFar = 50$



45