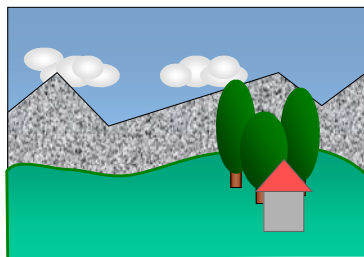


Rimozione delle superfici nascoste

- ✓ Soluzione 1: basata su ordinamento
 - ⇒ le primitive rasterizzate sovrascrivono nel frame buffer quelle rasterizzate in precedenza
 - ⇒ quindi, basta rasterizzare le primitive nell'*ordine giusto*
 - ordine back-to-front
 - ⇒ noto come "algoritmo del pittore"



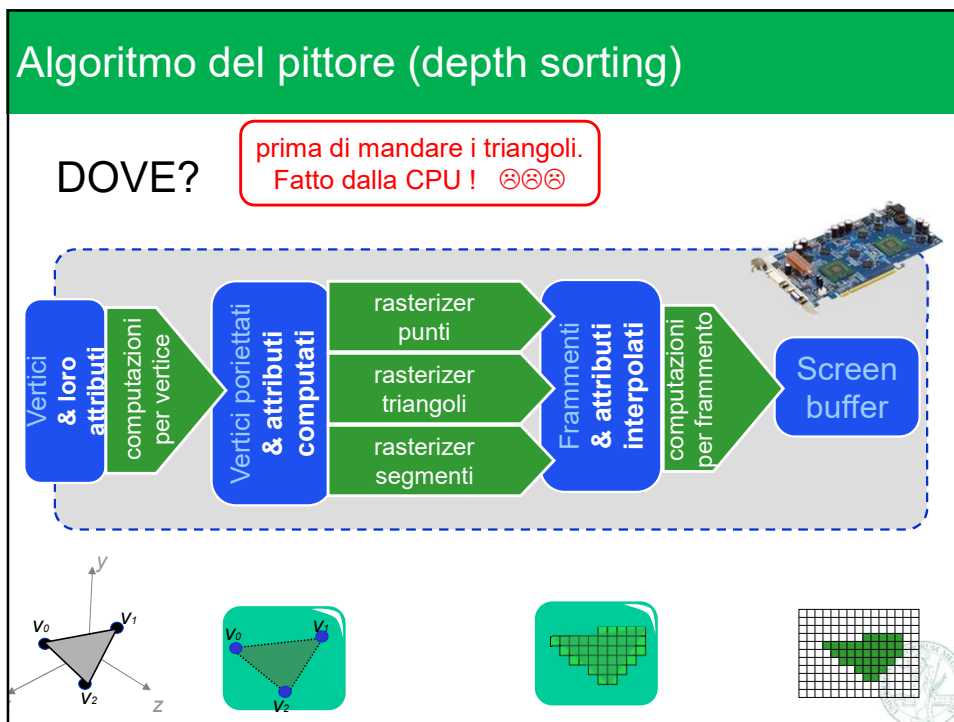
3

Algoritmo del pittore (depth sorting)

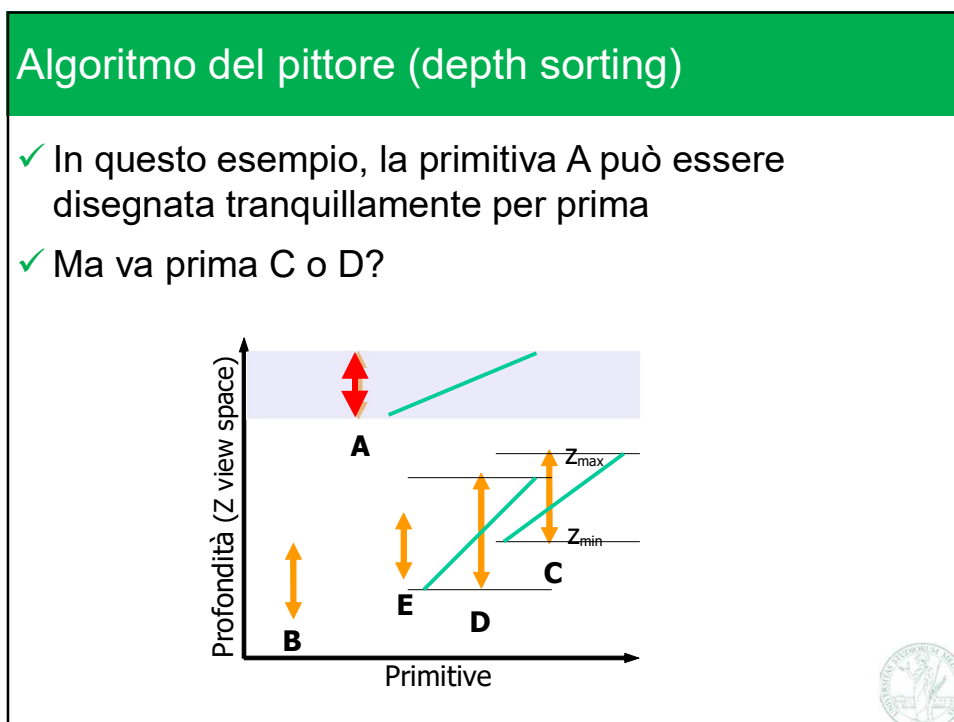
- ✓ Data una scena (composta da primitive)
 - ⇒ ordinare le primitive back-to-front
 - cioè in ordine di crescente di z... in spazio vista!
 - ⇒ rasterizzarle in successione



4



5

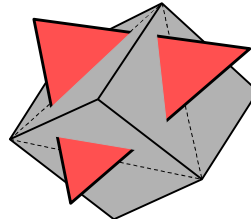


6

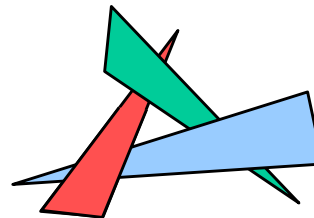
Algoritmo del pittore (depth sorting)

✓ Esiste sempre un ordine corretto?

⇒ NO.
Controesempio:
intersezioni



⇒ NO.
Controesempio:
cicli



7

Rimozione delle superfici nascoste tramite ordinamento: sommario degli ostacoli

✓ Limiti delle soluzioni basate su ordinamento:

- ⇒ Problema 1: ordinamento costa $O(n \log n)$ ("pseudolineare")
 - con n numero di primitive
 - n può molto grande. Es: se $n = \text{milioni}$ → $\log(n)$ è un fattore 30
In CG, peggio 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 (fino allo spazio vista)
- ⇒ 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 da i risultati corretti
- ⇒ Problema 4: complica il rendering
 - prescrivendo un certo ordine



8

Algoritmo del pittore (depth sorting)

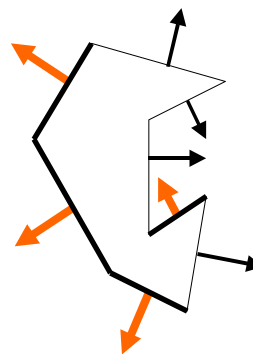
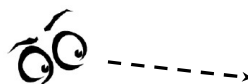
- ✓ Gli ostacoli sono superabili, ma l'algoritmo del pittore in pratica è usato raramente
 - ⇒ solo nelle occasioni in cui sia facile ordinare preventivamente le primitive back-to-front in fase di preprocessing
 - ⇒ esempio: terreno come campo di altezza
- ✓ Vediamo invece alternative per rimuovere le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



9

Caso particolare per mesh chiuse e ben orientate. Backface Culling

- ✓ Un caso particolare di Occlusion Culling
- ✓ Idea:
 - ⇒ ipotesi: superficie **chiusa** e **opaca**...
 - non vedrò mai l'interno!



10

Caso particolare: mesh chiuse e ben orientate

✓ Back-face Culling

⇒ “culling” (di una primitiva) =
scarto della primitiva in fase di rendering

✓ Idea:

⇒ ipotesi: superficie **chiusa**, **opaca** e **ben orientata**

→ non vedrò mai l'interno!

→ qualunque faccia si veda “da dietro”

(**back-facing triangles**)

finirà per forza *occlusa* alla vista da (almeno)
un layer di facce “viste da davanti”

(**front-facing triangles**)

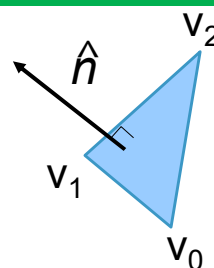
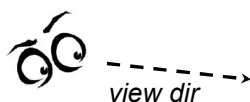
→ posso scartare tutte i back-facing triangles



11

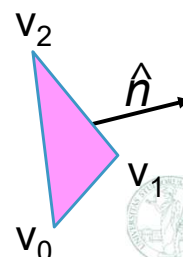
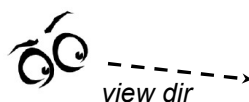
Caso particolare: mesh chiuse e ben orientate

✓ Traingolo *front-facing*

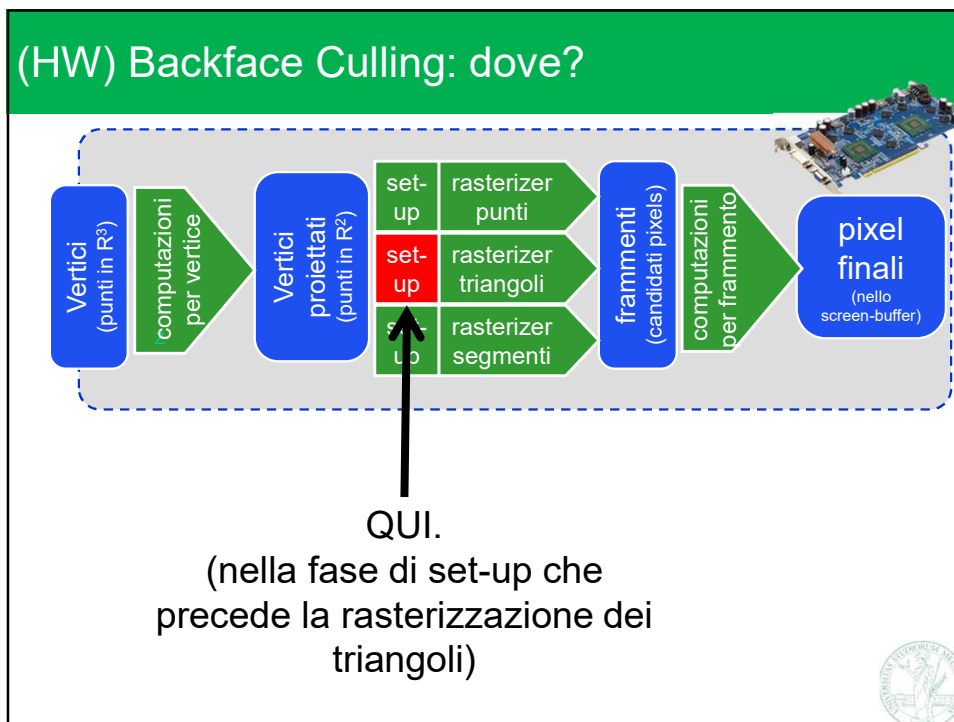


✓ Traingolo *back-facing*

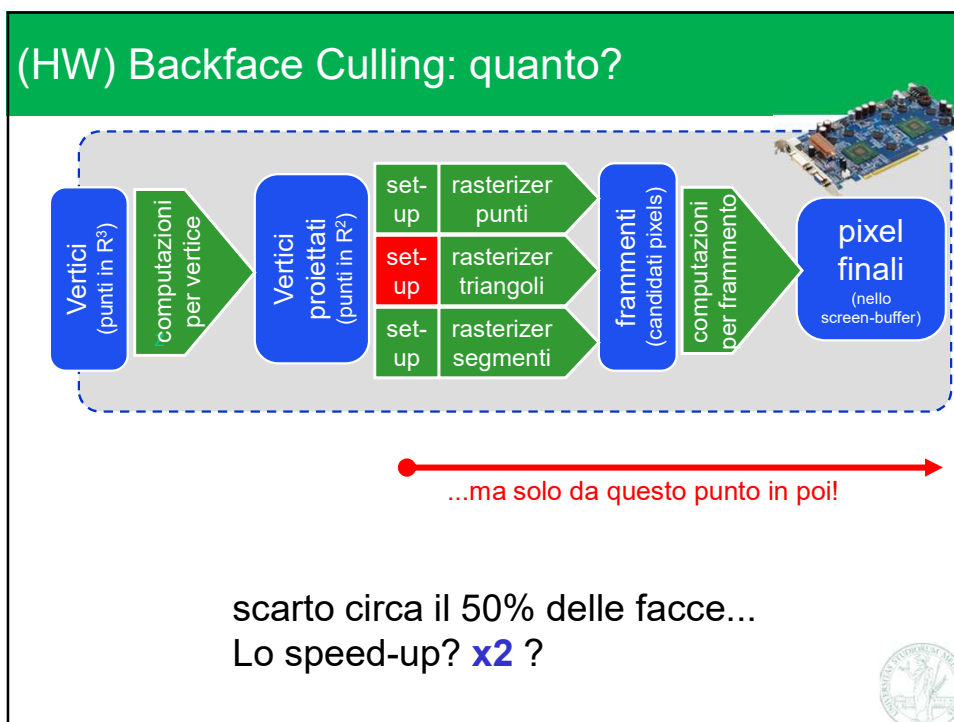
⇒ “visto di spalle”



14



15

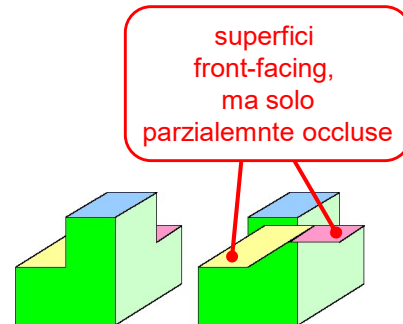


16

Rimozione delle superfici nascoste

✓ Back-face culling non basta

⇒ esempio:



20

Rimozione delle superfici nascoste con Depth-buffer

✓ Idea base: eseguire il test a livello di frammento

⇒ Di tutti i frammenti che insistono su un pixel, tengo solo quello di profondità (depth) minore

- cioè quello più vicino all'osservatore

⇒ Serve una struttura per memorizzare la profondità attuale di ogni pixel...

✓ «Depth-buffer»: (a volte: «Z-buffer»)

⇒ un buffer 2D

⇒ stessa risoluzione dello screen buffer

⇒ per ogni posizione [x , y] memorizza il depth del frammento attualmente presente in quella locazione

21

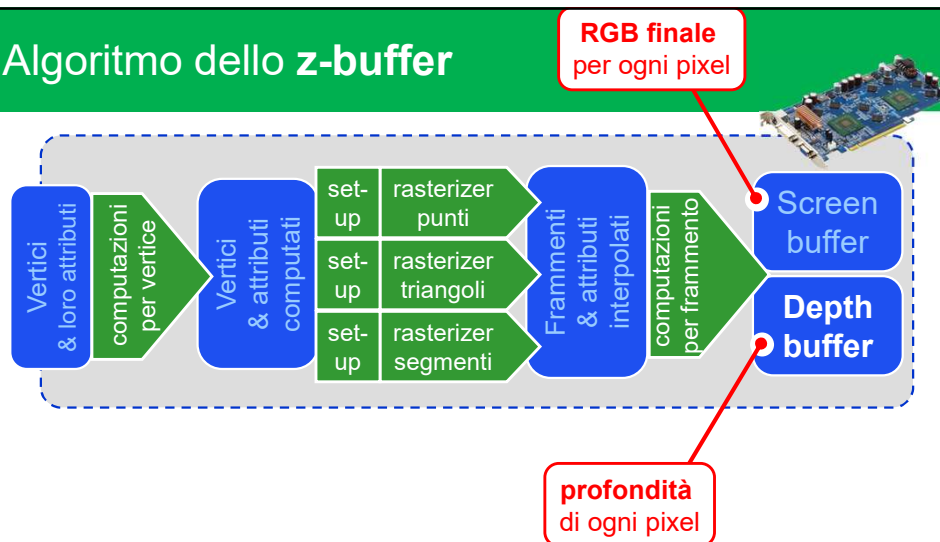
Valore di depth (profondità) di un frammento

- ✓ Un valore da 0 a 1
 - ⇒ 0: pixel più vicino possibile
 - ⇒ 1: pixel più lontano possibile
- ✓ Calcolato (per vertice) a partire dalla Z dello spazio clip
 - ⇒ Se $Z = -1 \rightarrow \text{depth} = 0$
 - ⇒ Se $Z = +1 \rightarrow \text{depth} = 1$
 - ⇒ $\text{depth} = (Z+1) / 2$
- ✓ Interpolato dentro ogni primitiva per ottenere il valore dei frammenti
 - ⇒ Come qualsiasi altro valore "varying" definite sui vertici
 - ⇒ Coordinate baricentriche

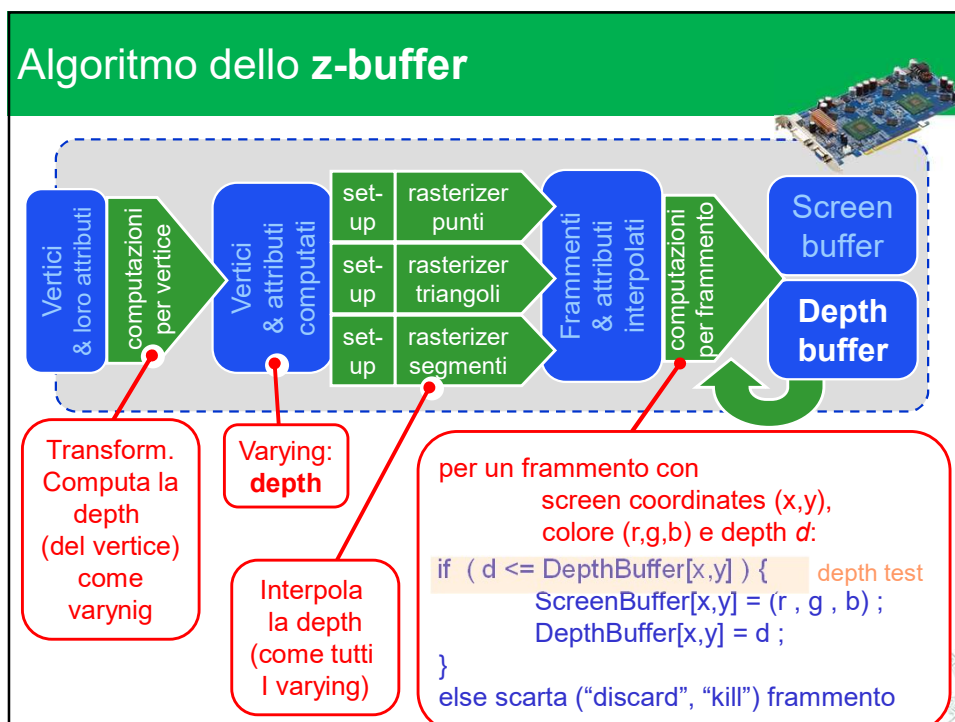


22

Algoritmo dello z-buffer



23



24

Rappresentazione reali in virgola fissa

- ✓ Rappresento i numeri reali in un intervallo predefinito (es: da 0 a 1) come interi
- ✓ Es: se ho 32 bit, allora, memorizzo l'intero x (da 0 a $2^{32}-1$) per rappresentare il razionale $x / (2^{32}-1)$
- ✓ Rispetto alla virgola mobile:
 - ⇒ intervallo limitato ☹️
 - ⇒ niente esponente! solo mantissa 😊
 - ⇒ distribuzione uniforme dei valori rappresentabili 😊☹️

25

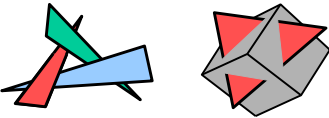
Algoritmo dello z-buffer: esempio

<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></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> <tr><td></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																=	<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>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> <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	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																																																																																																																																																																																													
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	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	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																																																																																																																																																																																													
<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>1.0</td><td></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></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></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></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></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></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></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	1.0		.5	.5	.5	.5	.5	1.0	1.0		.5	.5	.5	.5	1.0	1.0	1.0		.5	.5	.5	1.0	1.0	1.0	1.0		.5	.5	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	+	<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>1.0</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>.7</td><td>1.0</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	1.0	1.0	1.0	1.0	1.0	.4	.5	.7	1.0	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	1.0																																																																																																																																																																																														
.5	.5	.5	.5	.5	1.0	1.0																																																																																																																																																																																														
.5	.5	.5	.5	1.0	1.0	1.0																																																																																																																																																																																														
.5	.5	.5	1.0	1.0	1.0	1.0																																																																																																																																																																																														
.5	.5	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																																																																																																																																																																																													
.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	1.0	1.0	1.0	1.0	1.0																																																																																																																																																																																													
.4	.5	.7	1.0	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																																																																																																																																																																																													

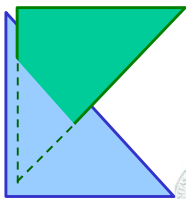
26

Algoritmo dello z-buffer: vantaggi

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


- ✓ Adatto all'implementazione parallela quindi HW 😊

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



27

Algoritmo dello z-buffer: limiti

- ✓ Costa un po' di memoria (GPU)
 - ⇒ spesso: $\text{sizeof}(\text{depth buffer}) = \text{sizeof}(\text{screen buffer}) / 2$
 - ⇒ il buffer deve essere inizializzato (a profondità massima) prima di ogni rendering
- ✓ Problemi di *aliasing sulla z*
 - ⇒ causa detto: "z-fighting"
 - ⇒ quando la precisione non è sufficiente
es., se si renderizzano due superfici parallele molto vicine
- ✓ I frammenti vengono scartati solo alla fine del pipeline
 - ⇒ tutta la computazione è già stata inutilmente effettuata
- ✓ Assume superfici del tutto *opache*
 - ⇒ problemi con le superfici *semitrasparenti*
- ✓ Dati condivisi in lettura e scrittura ☹
 - ⇒ complicazione per chi implementa HW
 - ⇒ efficienza ameno in parte impattata (ma test molto ottimizzato)
 - ⇒ è un test **HardWired in GPU** (vediamo come)



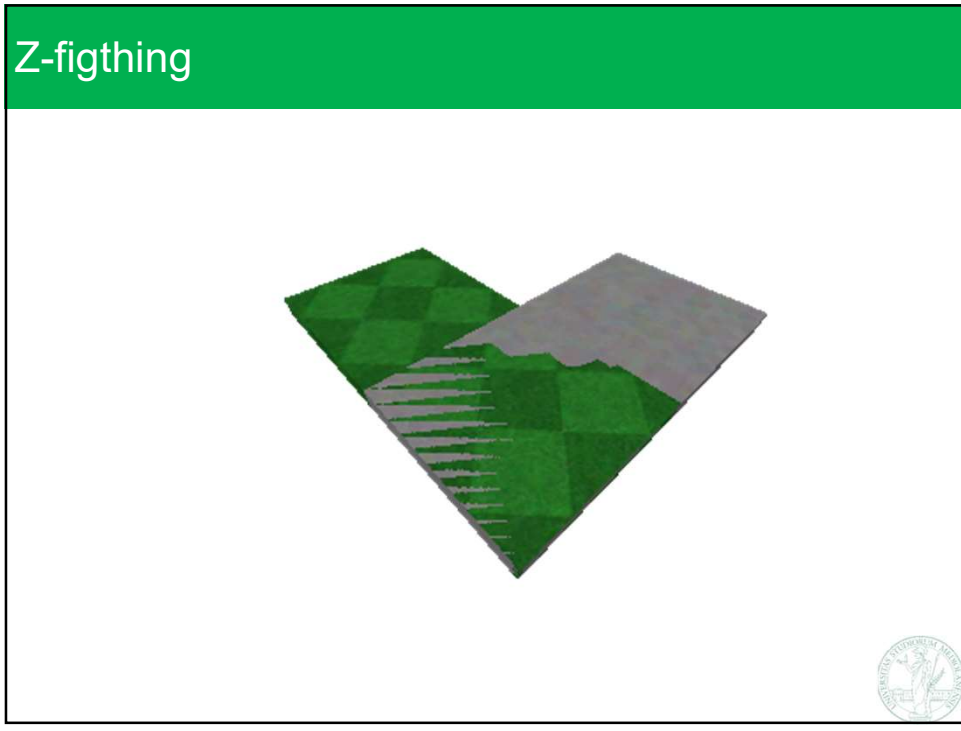
28

Algoritmo dello z-buffer

- ✓ Il rendering è order independent:
 - ⇒ se disegno prima la primitiva davanti:
quella dietro verrà scartata
 - ⇒ se disegno prima la primitiva dietro:
quella davanti la sovrascrive
 - ⇒ risultato finale: lo stesso!
 - ⇒ (ma l'efficienza può variare)



29



30



31

Il rendering produce anche depth images

- ✓ Effetto collaterale dell'algoritmo di depth buffer: dopo il rendering, ho prodotto non solo un'immagine (RGB per pixel) ma anche un depth-buffer (profondità per pixel)
 - ⇒ una depth image con un «bassorilievo» della scena renderizzata



- ⇒ Il buffer RGB viene mandato a schermo,
- ⇒ il depth-buffer di solito viene semplicemente scartato
- ⇒ ...ma alcuni algoritmi lo sfruttano invece per gli scopi più vari



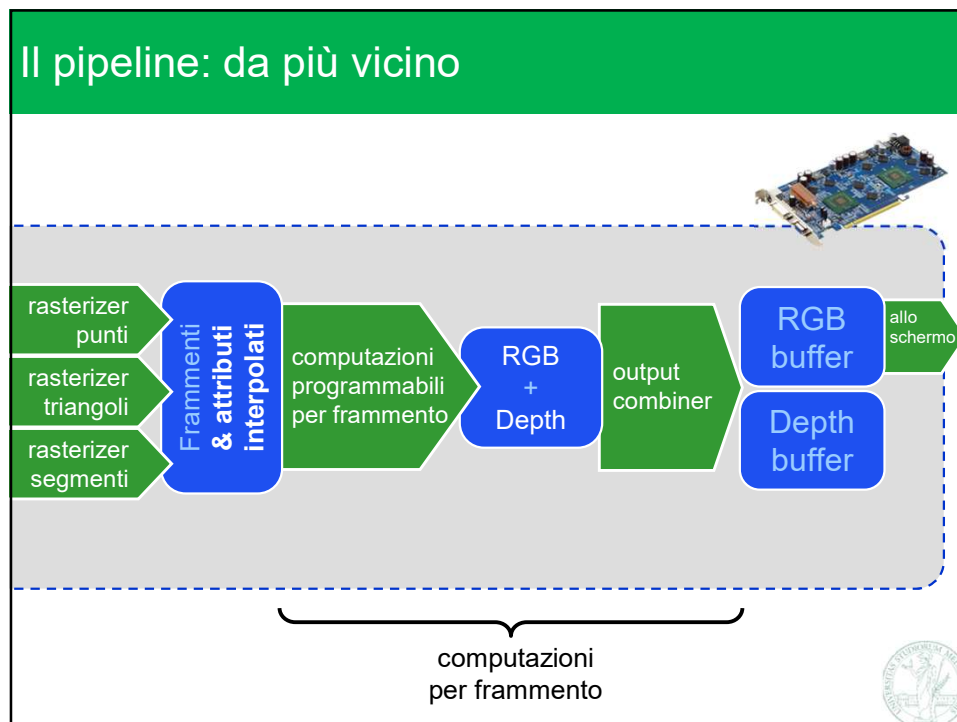
32

Depth test: considerazioni sull'efficienza

- ✓ Il depth test è hard-wired affinché sua implementazione sia ottimizzata in GPU
 - ⇒ quindi nell'output combiner, non nel fragment shader programmabile!
 - ⇒ necessario, perché il depth-test richiede lettura + scrittura in memoria condivisa (il Depth Buffer): problema per il parallelismo!
- ✓ Molte ulteriori ottimizzazioni sono fatte in HW (silenziosamente) per consentire
 - ⇒ di anticipare il test (**early depth test**) e/o
 - regola generale per i test che scartano: prima vengono fatti meglio è
 - ⇒ di testare gruppi di frammenti alla volta, o intere primitive (**hierarchical Z buffer**)



33



34

Depth test: considerazioni pratiche sull'efficeinza

- ✓ scartare è più efficiente di scrivere-per-poi-sovrascrivere
 - ⇒ quindi: meglio disegnare *prima* le cose davanti *poi* quelle dietro: **rendering front-to-back**
 - ⇒ l'impatto è accentuato dalle ottimizzazioni HW
 - ⇒ ironia: il depth-sorting è ancora utile!
 - ⇒ ma, non è necessario (il risultato è cmq corretto)
 - es: può agevolmente essere fatto a livello di oggetto, non di primitive
- ✓ cambiare il depth di default (dal fragment shader) costringe l'HW ad eseguire il test *davvero* all'ultimo
 - ⇒ ciò disabilita le ottimizzazioni HW accennate sopra: potenzialmente grosso impatto sulle performances

35

Per completezza: altri per-fragment tests effettuati nell'output combiner

Il depth test è solo uno dei test hardwired che possono essere abilitati nell'output combiner.

Altri sono:

✓ *Scissor test*

scarta i frammenti esterni ad un dato rettangolo

✓ *Stencil test*

scarta i frammenti usando una maschera arbitraria



36

Considerazioni pratiche sulla scelta di zNear e zFar

- ✓ a volte è utile scartare il troppo vicino (con **zNear**)
 - ⇒ es, telecamera «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 (con **zFar**)
 - ⇒ gli oggetti appaiono più piccoli con la distanza, ma mai fino a sparire
 - ⇒ perché ometterli del tutto?
 - ⇒ dunque, idealmente, $zFar = +\infty$ ma...
- ✓ entrambe le cose sono comunque necessarie
 - ⇒ zFar grande, o zNear piccolo → grande intervallo di Z vista mappato nel solito intervallo in Z clip → precisione numerica si riduce → z Fighting ☹
- ✓ nota: **zFar > zNear > 0** sempre (altrimenti: divisioni per zero!)
- ✓ esempio di scelta ragionevole:
«inquadro da 20 cm a 100 metri» → $zNear = 0.2$, $zFar = 100$



38