

Grafica Computazionale

Blending

Fabio Ganovelli

fabio.ganovelli@isti.cnr.it

a.a. 2005-2006

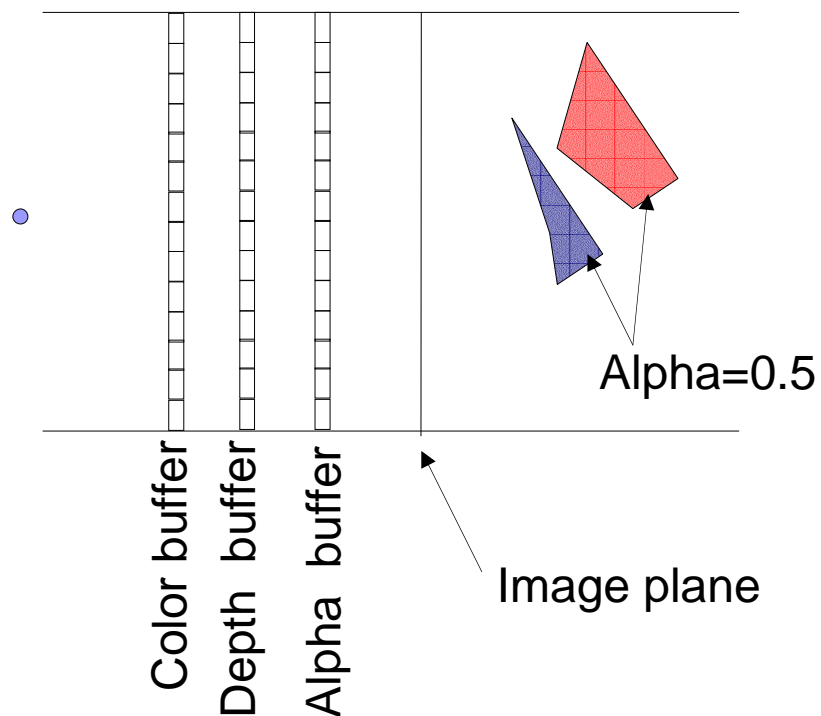


Il canale alpha

- Fino ad ora abbiamo sempre assunto che tutte le superfici fossero perfettamente opache == la luce non ci passa attraverso
- Come si possono visualizzare oggetti semi trasparenti?
- Si aggiunge una quarta componente al colore (oltre Red, Green, Blue: l'opacità, convenzionalmente riferita con Alpha.
 - Alpha =0: completamente trasparente
 - Alpha =1: completamente opaco
- Durante la rasterizzazione, il valore scritto nel pixel non è più solo r,g,b ma anche alpha. Ne segue che:
- Oltre al color buffer e al depth buffer (z-buffer) ci serve anche un alpha buffer



Blending: composizione del colore



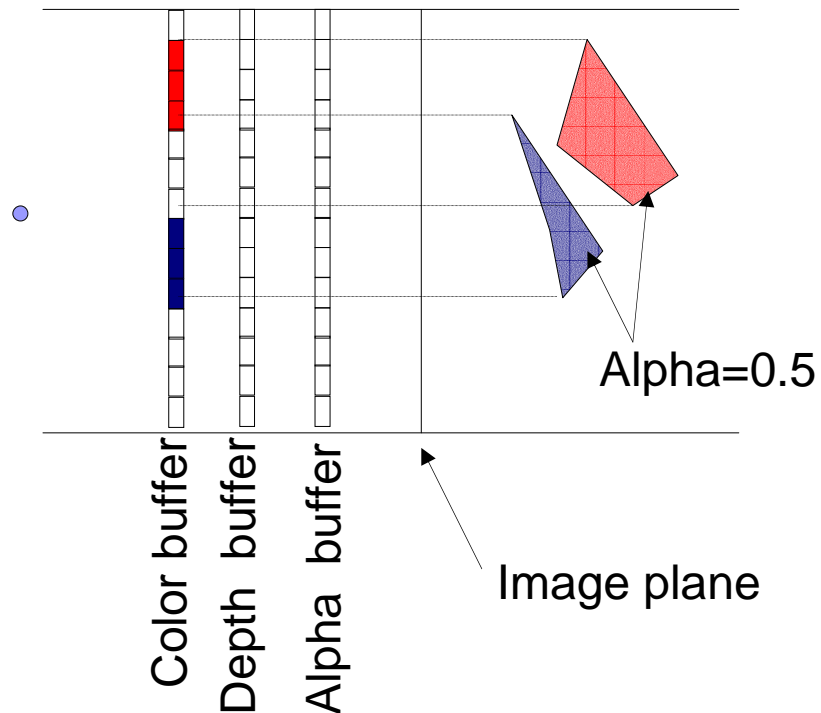
Dimentichiamoci per un attimo dello z-buffering

Di che colore facciamo i pixel accesi dal filling di entrambi i poligoni?

tenendo presente che i due poligoni vengono rasterizzati in maniera indipendente



Blending: composizione del colore



Funziona come il depth test:

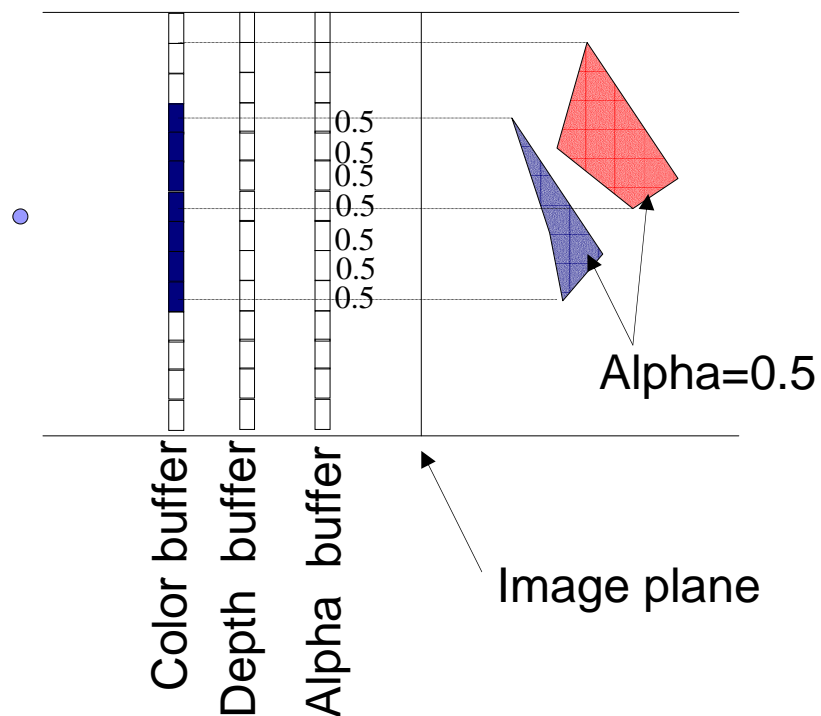
Durante la rasterizzazione, oltre al colore viene scritto il valore di opacità nell'alpha buffer.

Il colore è determinato da:

- colore e alpha presenti nel buffer
- colore e alpha del frammento che si sta rasterizzando



Blending: Esempio



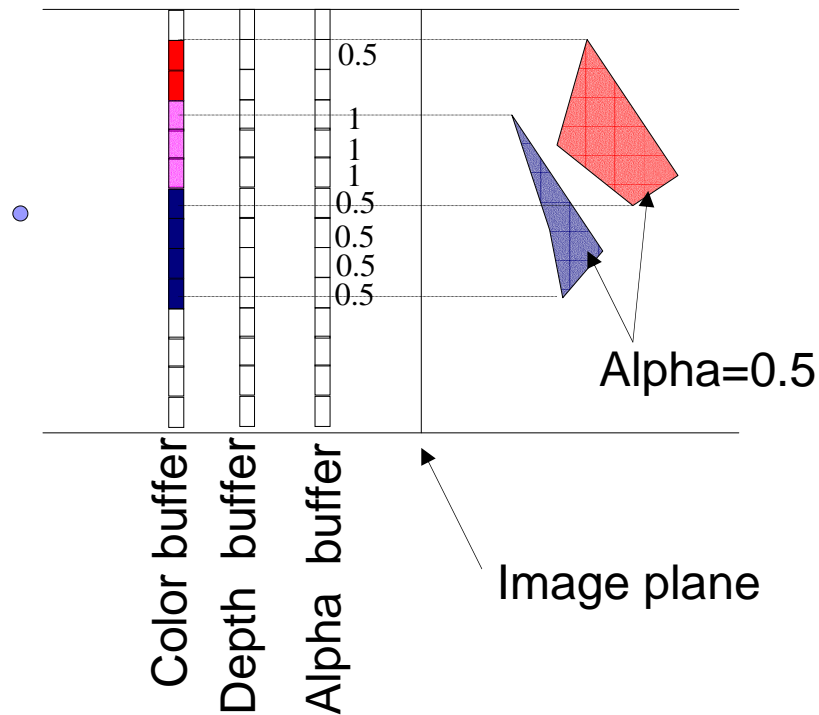
Supponiamo che venga rasterizzato prima il poligono blu e poi il rosso.

Stato dei buffer dopo la rasterizzazione del poligono blu.

Notate che il colore nel color buffer è saturo...questo perché l'opacità viene scritta sul canale alpha



Blending: Esempio



Supponiamo che venga rasterizzato prima il poligono blu e poi il rosso.

Stato dei buffer dopo la rasterizzazione del poligono rosso.

I valori sono stati aggiornati con l'equazione:

$$\begin{pmatrix} r \\ g \\ b \\ \alpha \end{pmatrix}_d = \begin{pmatrix} r \\ g \\ b \\ \alpha \end{pmatrix}_d \cdot \begin{pmatrix} s_r \\ s_g \\ s_b \\ s_\alpha \end{pmatrix} + \begin{pmatrix} r \\ g \\ b \\ \alpha \end{pmatrix}_s \cdot \begin{pmatrix} d_r \\ d_g \\ d_b \\ d_\alpha \end{pmatrix}$$

Nuovo valore

Valore attuale

Fattori decisi dall'utente

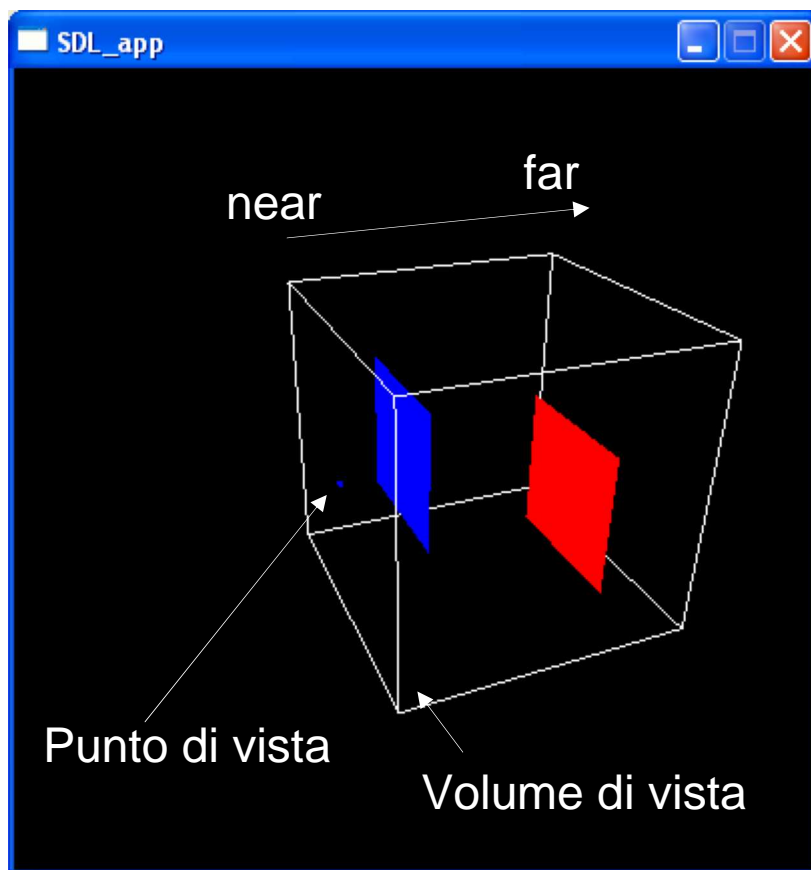
Valore del frammento in produzione

Con:

$$\begin{pmatrix} s_r \\ s_g \\ s_b \\ s_\alpha \end{pmatrix} = \begin{pmatrix} d_r \\ d_g \\ d_b \\ d_\alpha \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$



Blending: un esempio



Supponiamo di visualizzare la scena dal punto di vista specificato

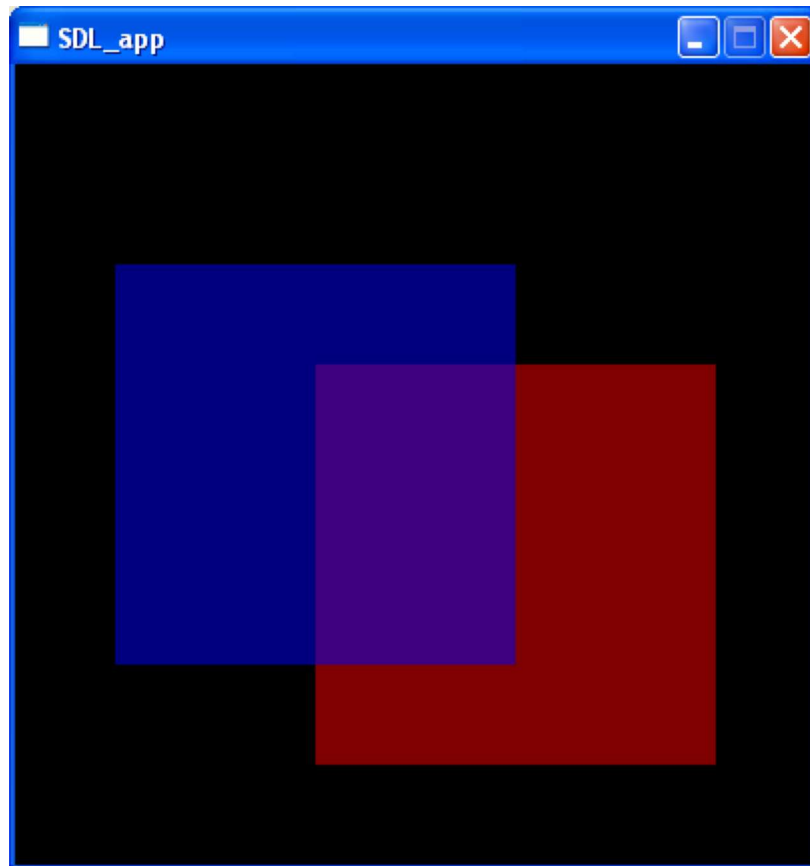
Vediamo qualche esempio al variare dei fattori della combinazione lineare

Il rendering del poligono più lontano viene fatto per primo

$\alpha = 0.5$ per entrambi i poligoni



Blending: esempi

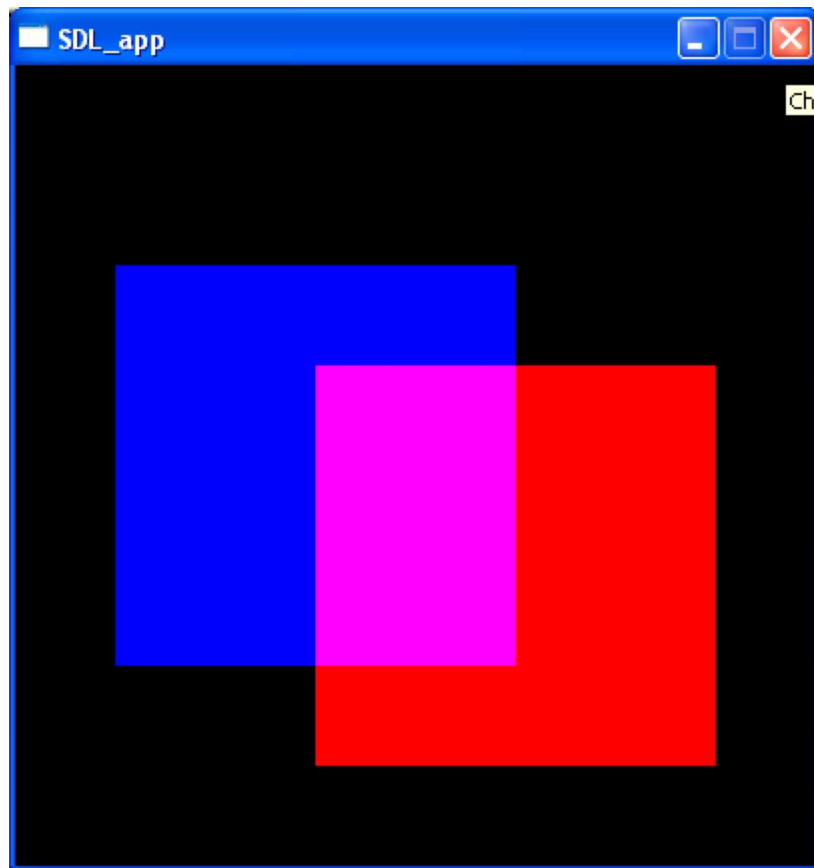


$$\begin{pmatrix} r \\ g \\ b \\ \alpha \end{pmatrix}_d = \begin{pmatrix} r \\ g \\ b \\ \alpha \end{pmatrix}_d \cdot \begin{pmatrix} s_r \\ s_g \\ s_b \\ s_\alpha \end{pmatrix} + \begin{pmatrix} r \\ g \\ b \\ \alpha \end{pmatrix}_s \cdot \begin{pmatrix} d_r \\ d_g \\ d_b \\ d_\alpha \end{pmatrix}$$

$$\begin{matrix} s_r & s_g & s_b & s_\alpha & d_r & d_g & d_b & d_\alpha \\ \alpha_s & \alpha_s & \alpha_s & \alpha_s & 1-\alpha_s & 1-\alpha_s & 1-\alpha_s & 1-\alpha_s \end{matrix}$$



Blending: esempi

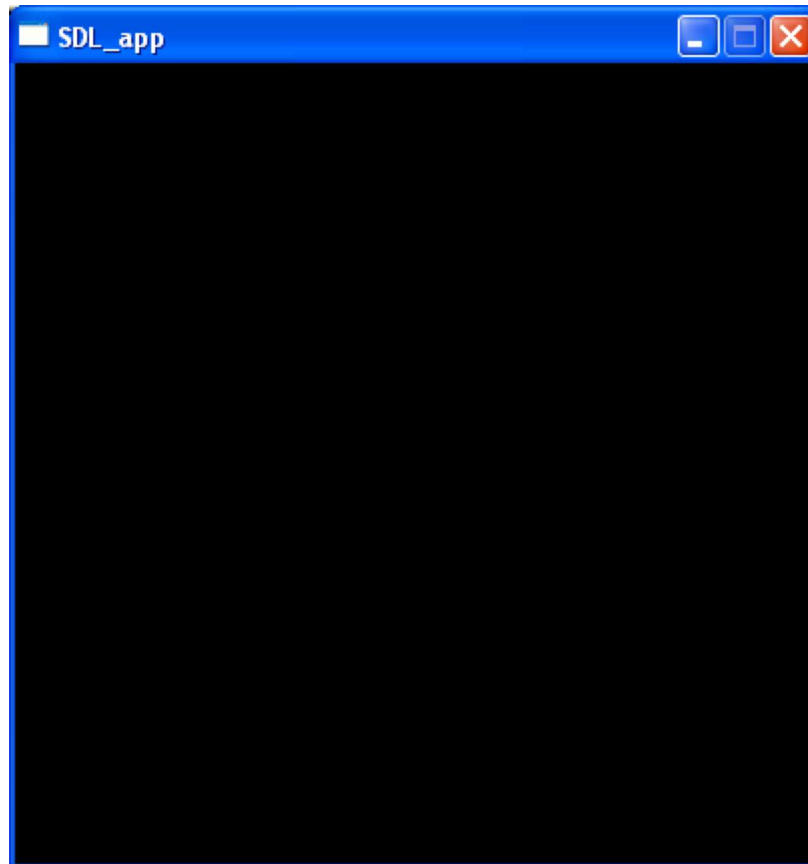


$$\begin{pmatrix} r \\ g \\ b \\ \alpha \end{pmatrix}_d = \begin{pmatrix} r \\ g \\ b \\ \alpha \end{pmatrix}_d \cdot \begin{pmatrix} s_r \\ s_g \\ s_b \\ s_\alpha \end{pmatrix} + \begin{pmatrix} r \\ g \\ b \\ \alpha \end{pmatrix}_s \cdot \begin{pmatrix} d_r \\ d_g \\ d_b \\ d_\alpha \end{pmatrix}$$

s_r	s_g	s_b	s_α	d_r	d_g	d_b	d_α
α_s	α_s	α_s	α_s	$1 - \alpha_s$	$1 - \alpha_s$	$1 - \alpha_s$	$1 - \alpha_s$
1	1	1	1	1	1	1	1



Blending: esempi

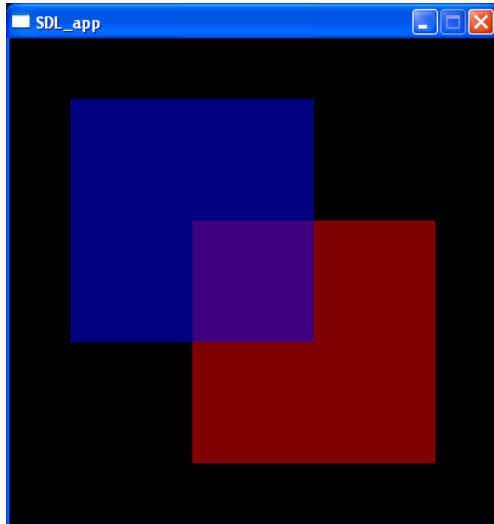


$$\begin{pmatrix} r \\ g \\ b \\ \alpha \end{pmatrix}_d = \begin{pmatrix} r \\ g \\ b \\ \alpha \end{pmatrix}_d \cdot \begin{pmatrix} s_r \\ s_g \\ s_b \\ s_\alpha \end{pmatrix} + \begin{pmatrix} r \\ g \\ b \\ \alpha \end{pmatrix}_s \cdot \begin{pmatrix} d_r \\ d_g \\ d_b \\ d_\alpha \end{pmatrix}$$

s_r	s_g	s_b	s_α	d_r	d_g	d_b	d_α
α_s	α_s	α_s	α_s	$1-\alpha_s$	$1-\alpha_s$	$1-\alpha_s$	$1-\alpha_s$
1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0

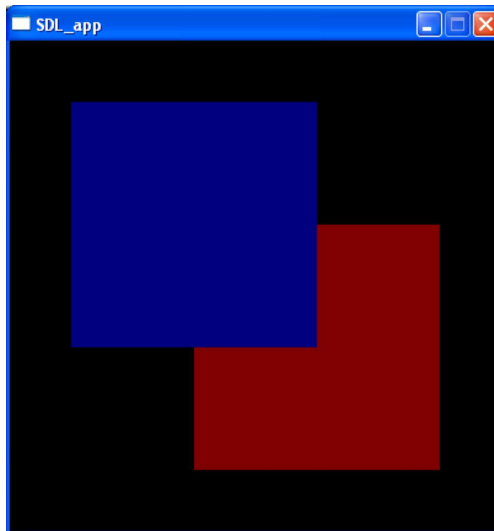


Blending e z-buffering: l'ordine conta



Disegno prima il rosso e poi il blu (“back to front”)

Quando rasterizzo il poligono blu trovo già il colore rosso con $\alpha = 0.5$ e il colore risultante è la composizione (coefficienti della prima riga)



Disegno prima il blu e poi il rosso (“front to back”)

Quando rasterizzo il poligono rosso, i frammenti coperti dal blu vengono scartati





Blending e z-buffering: l'ordine conta

- Se si usa il blending, occorre stare attenti all'ordine in cui si disegna
- Le opzioni più funzionali per implementare la trasparenza sono:
 - Disegnare prima le cose più lontane (algoritmo del pittore)
 - Usare i coefficienti nella prima linea della tabella
- Un po' di nomi:
 - Z Kill: scartare un frammento a causa della profondità
 - Alpha Kill: scartare un frammento perché la sua opacità è troppo piccola ("praticamente" trasparente)

