

# Shading

Un modello di illuminazione fornisce una funzione  $L(x, \text{luci}, \text{oggetti}) = \text{colore}$ . Lo shading è il modo in cui applichiamo  $L$  per calcolare il colore di ciascun frammento.

## Flat shading

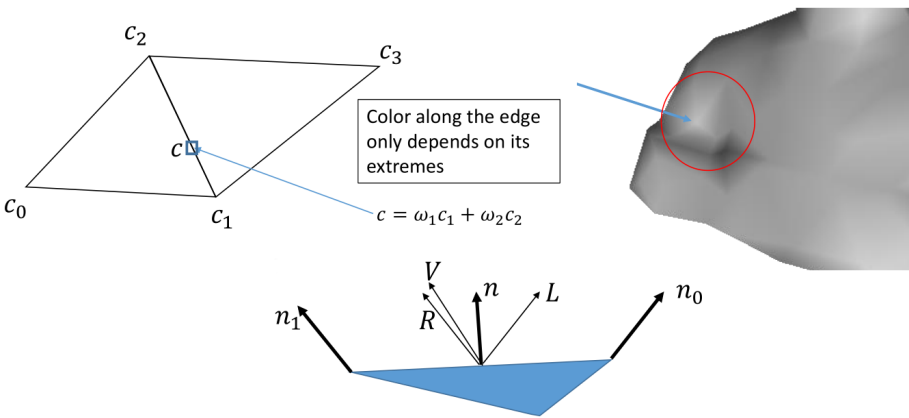
Calcolo per faccia, tutti i frammenti della faccia hanno lo stesso colore (sicuro?). Rende evidente la triangolazione a causa della discontinuità di colore tra facce adiacenti

Richiede di conoscere la normale della faccia. Potremmo passare la stessa normale per tutti i suoi vertici, ma dovremmo crearne copie per triangoli che li condividono, quindi la calcoliamo nel fragment shader:

```
N = normalize(cross(dFdx(pos), dFdy(pos)))
```

## Gouraud shading

Si calcola il colore per ogni vertice e si interpola il risultato dentro il triangolo. La qualità del risultato dipende dalla risoluzione della mesh:



Nell'ultimo esempio si perde un highlight speculare, nel caso in cui la superficie originale fosse arrotondata in quel punto.

## Phong shading

Si calcola l'illuminazione nel fragment shader, con la normale interpolata. Questa deve essere normalizzata, e il risultato non è sempre corretto (normali dei vertici a 180°), ma il costo di fare slerp non è giustificato dal piccolo miglioramento nella qualità.

	output vertex shader	fragment shader
Gauraud	posizione, colore	-
Phong	posizione, normale, direzione di vista, altro	calcola illuminazione