

Corso di Visione Artificiale

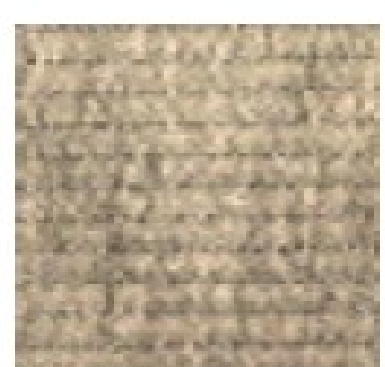
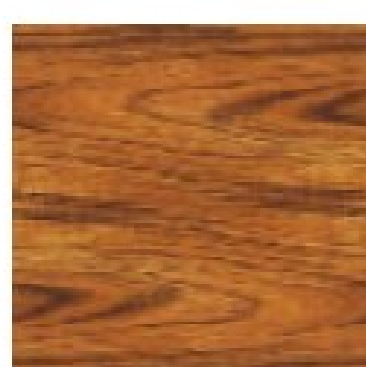
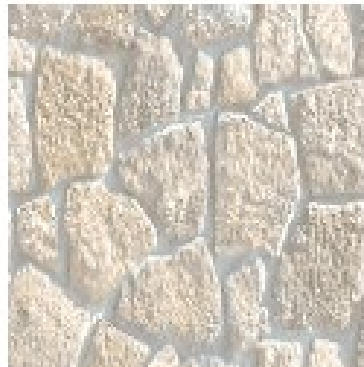
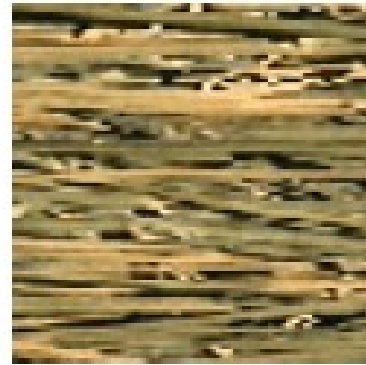
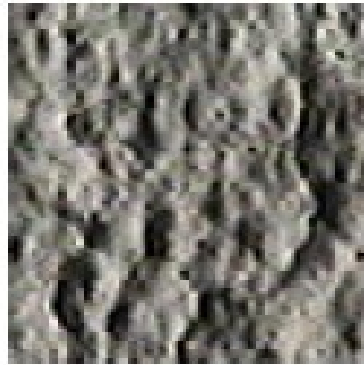
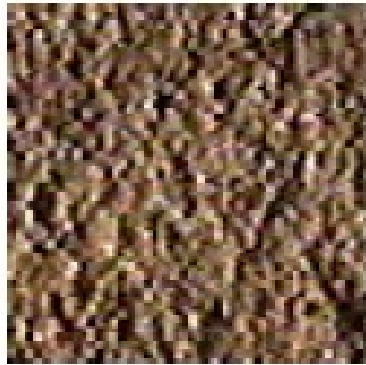
The background features a large, faint watermark of the seal of Ca' Foscari University of Venice. The seal is circular and contains a central figure, likely a lion or a similar heraldic symbol, surrounded by Latin text. The text includes "UNIVERSITATIS CA' FOSCARII" at the top and "IN DOMO FOSCARII" at the bottom. A central shield contains the motto "TIBI HOC MARLISTA CE HEUS".

Texture

Samuel Rota Bulò

Texture

- Le texture sono facili da riconoscere ma difficili da definire.



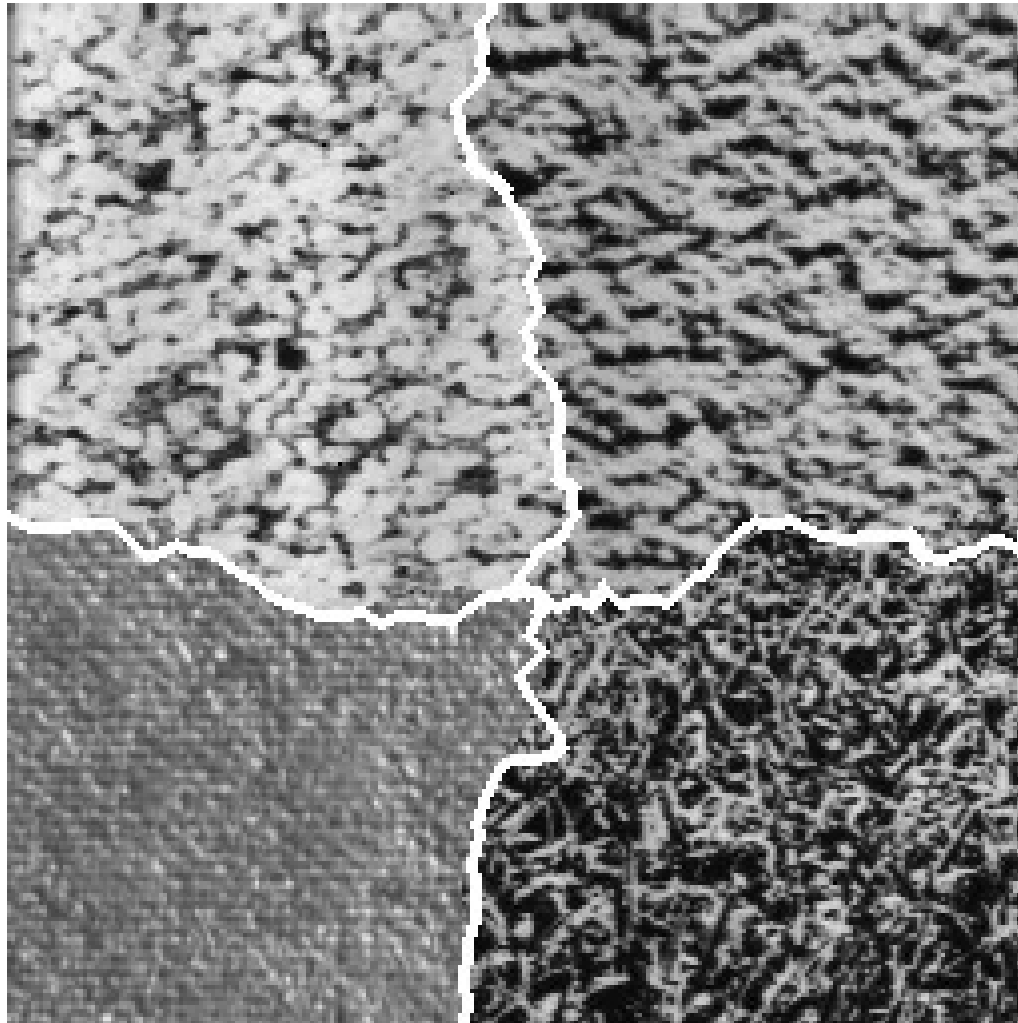
Texture

- Il fatto di essere una texture dipende dal livello di scala a cui si osserva.



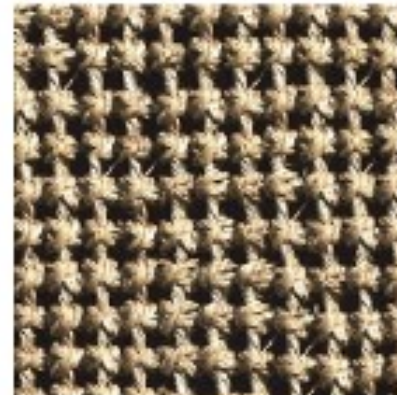
Applicazioni

- **Segmentazione di texture:** riconoscere regioni con texture uniforme.



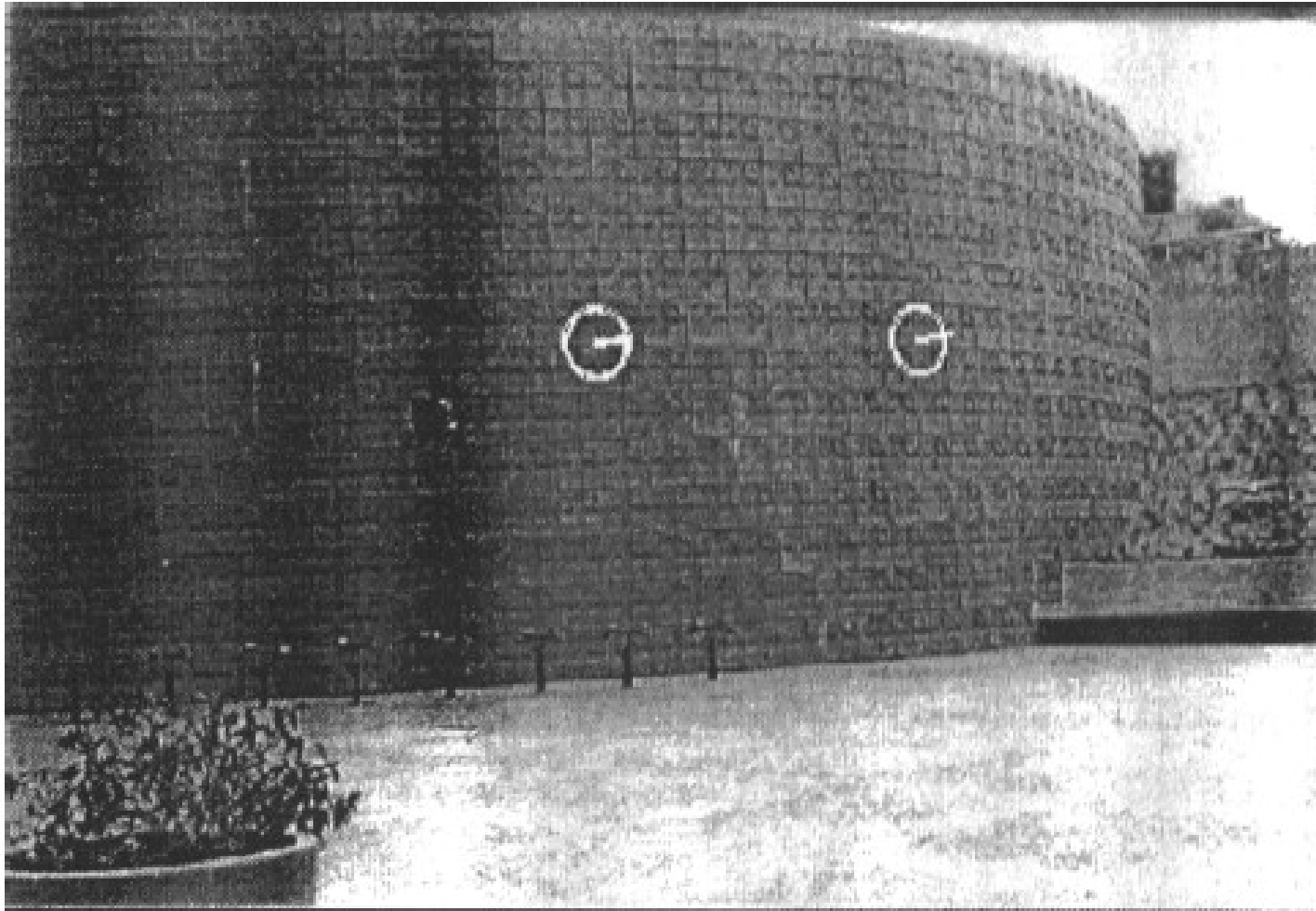
Applicazioni

- **Sintesi di texture:** ricreare regioni con la texture a partire da piccoli esempi.



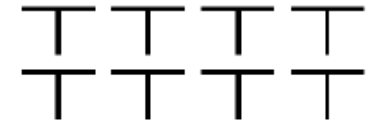
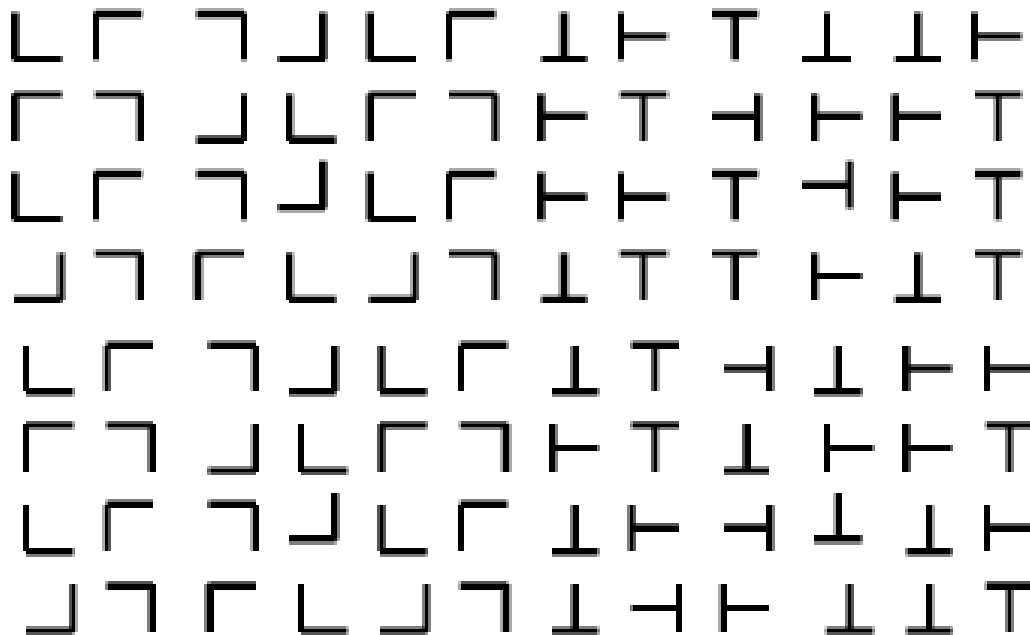
Applicazioni

- **Forma da texture:** riconoscere l'orientamento di una superficie o ricostruirla a partire da texture.



Textons

- Le texture tipicamente sono composte da pattern di elementi regolari detti **textons**.
- Un modo per descrivere una texture è quello di riconoscere i textons e descrivere come sono organizzati spazialmente.
- La difficoltà di questo approccio è che in generale non c'è un insieme canonico di textons da cercare.

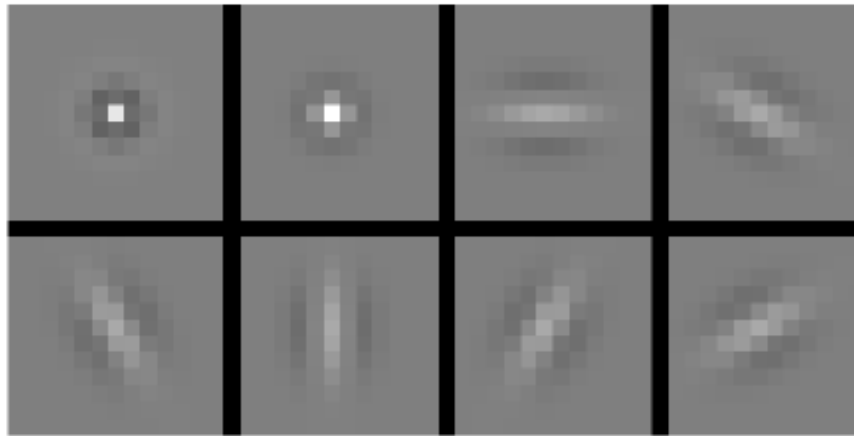


Rappresentazione di Texture

- Conviene cercare pattern di basso livello (punti e linee).
- Un modo per trovare questi pattern in una texture è, come abbiamo già visto, mediante filtraggio.
- Abbiamo una risposta forte del filtro se l'intorno dell'immagine è simile ad esso.
- Possiamo quindi descrivere la texture mediante una collezione di filtri.

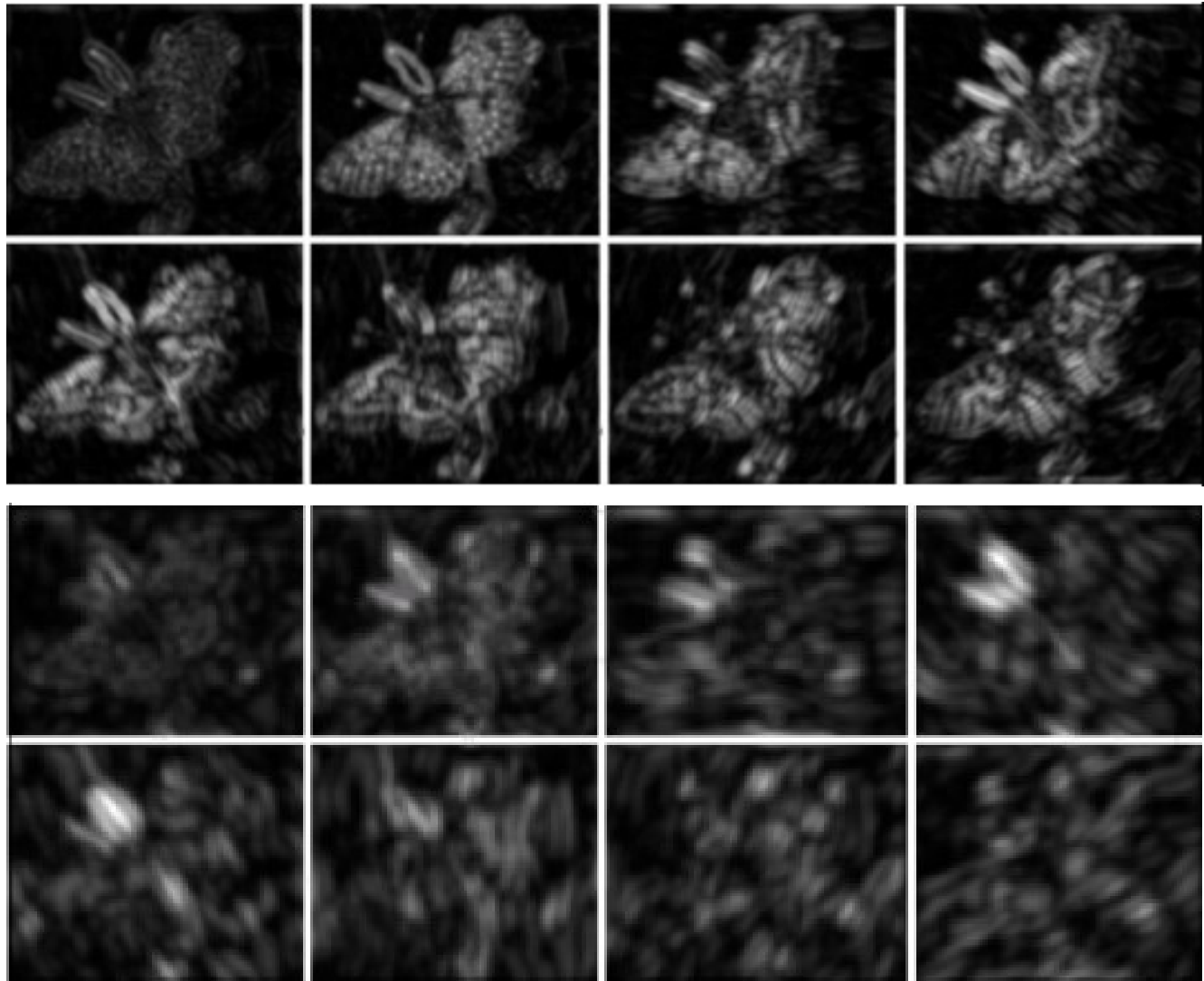
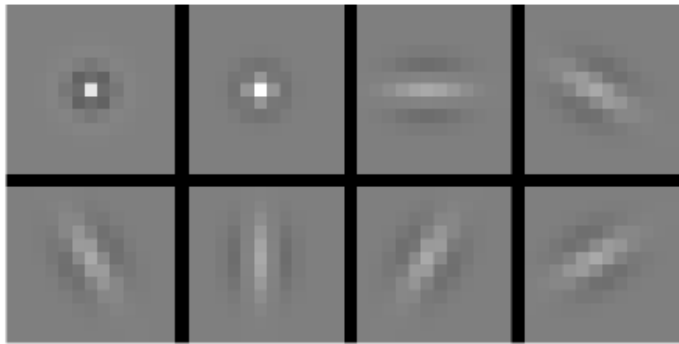
Filtri puntuali:
somma pesata di
2 Gaussiane
concentriche

Filtri puntuali:
somma pesata di
3 Gaussiane
concentriche



Filtri direzionali:
somma pesata di
3 Gaussiane
orientate sfasate

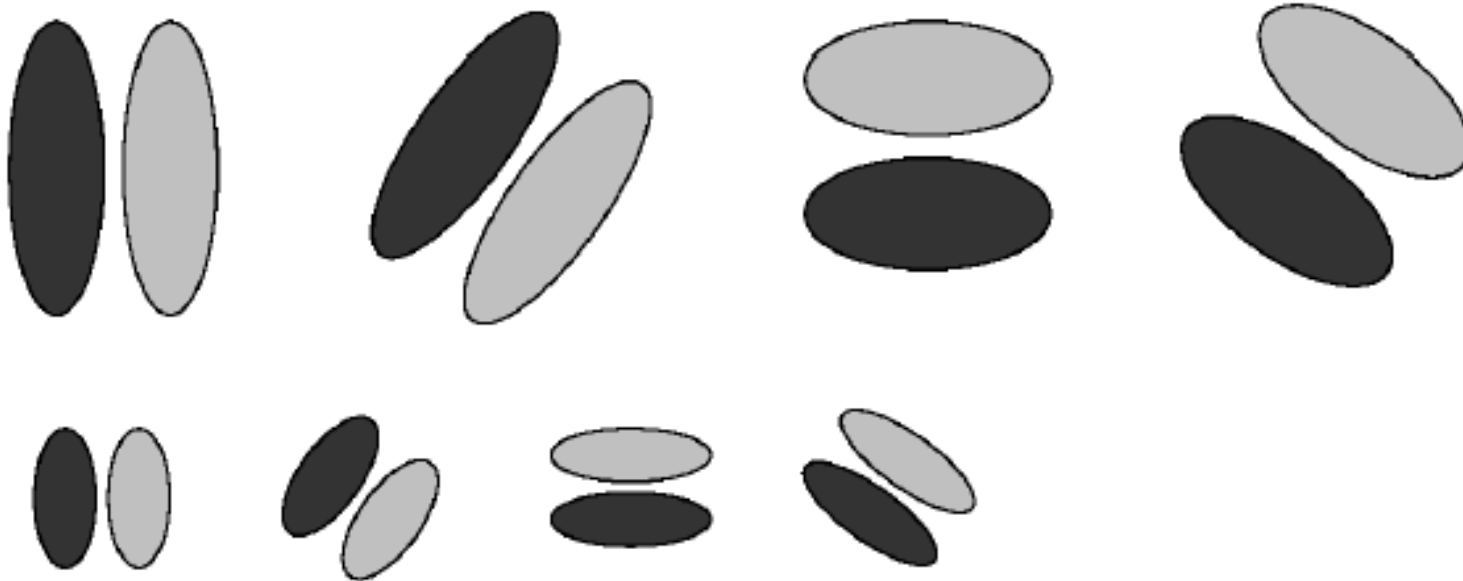
Rappresentazione di Texture



scala più
alta



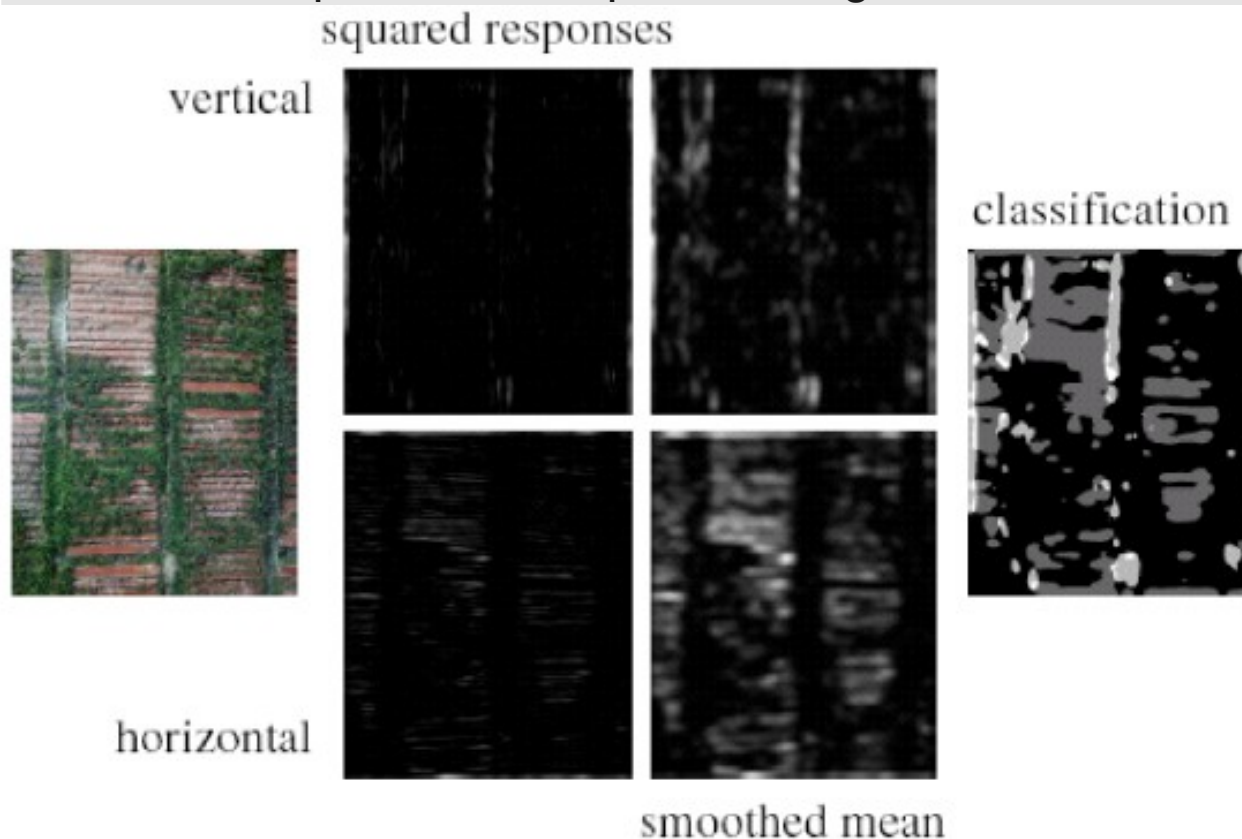
Quanti e che filtri?



- Non c'è una risposta canonica.
- In generale, non è necessario costruire filtri complicati, dal momento che combinazioni di filtri puntuali e orientati ottengono risultati simili.
- L'importante è creare molti filtri semplici (puntuali, direzionali) a scale, fasi e direzioni diverse.

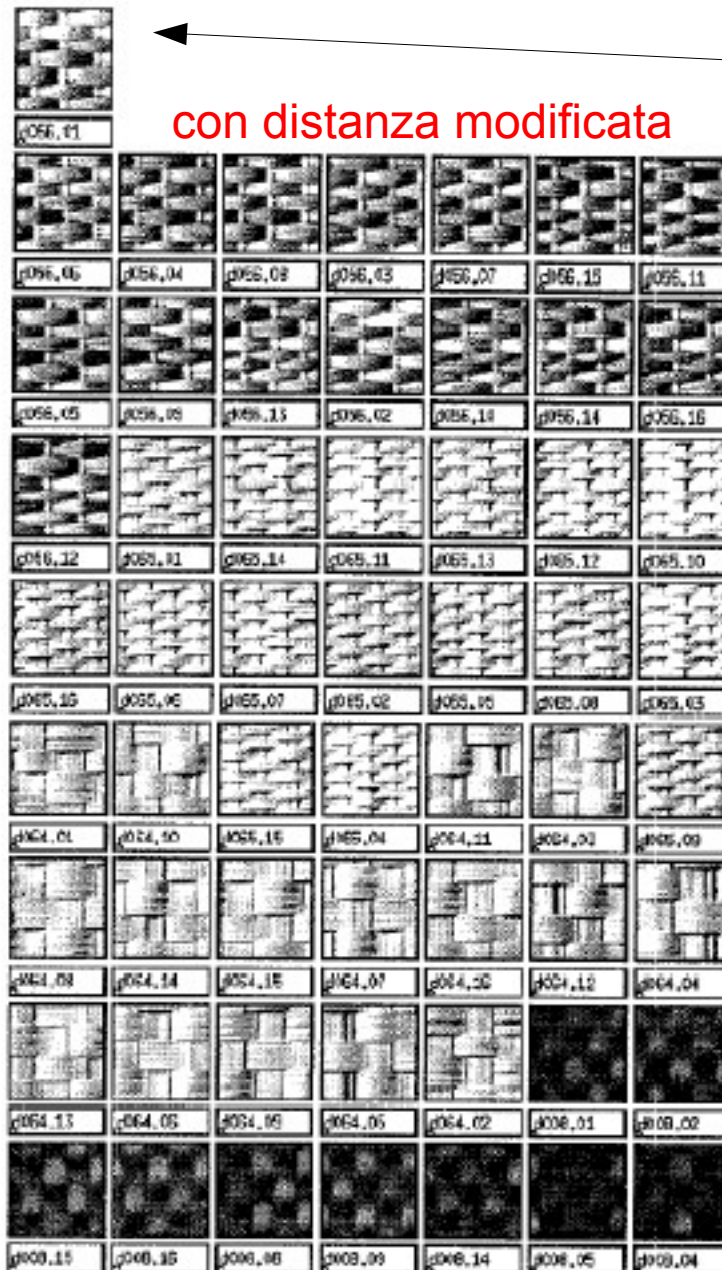
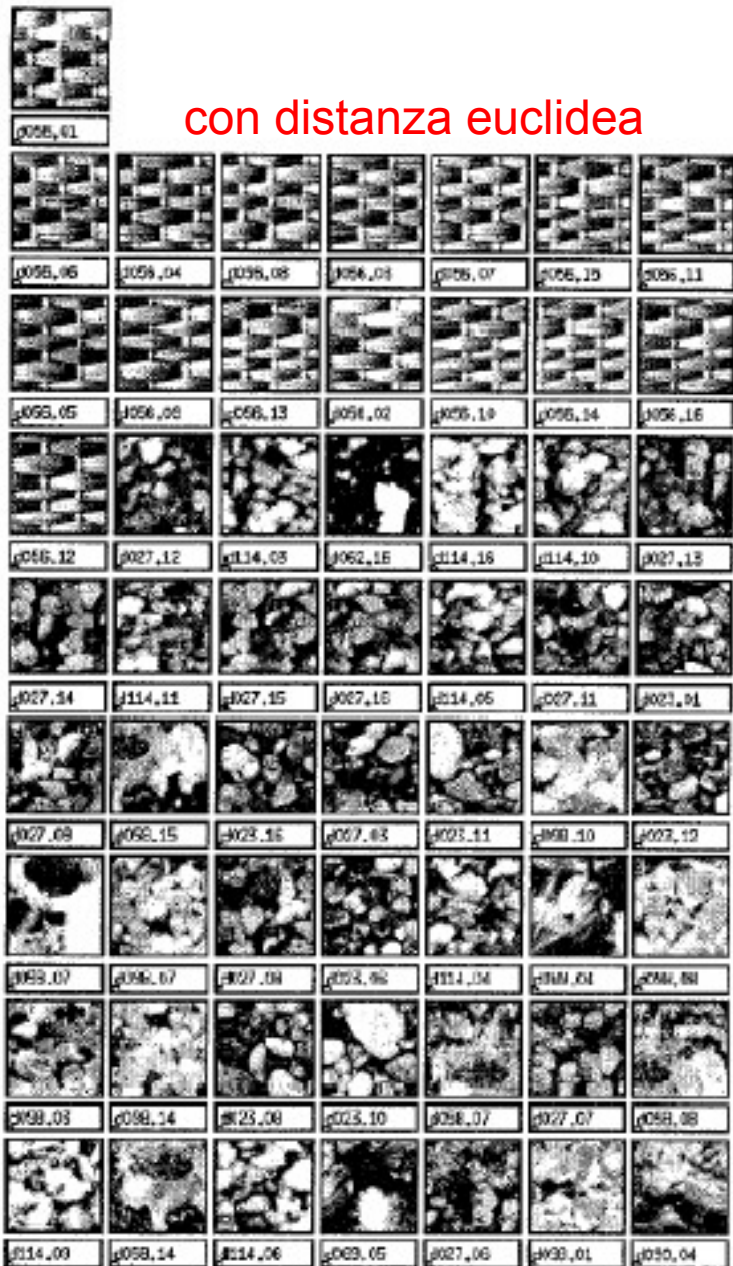
Rappresentazione di texture

- Avere una collezione di risposte di filtri non è di per sé una rappresentazione di texture.
- Immaginiamo di conoscere le dimensioni di della finestra in cui vogliamo rappresentare una texture. Una rappresentazione consiste in una collezione di statistiche delle risposte dei filtri in quella finestra.
- Un esempio è prendere la media delle risposte al quadrato per discriminare risposte forti/deboli piuttosto che positive/negative.



- Un altro è prendere la media e la deviazione standard delle risposte dei filtri all'interno della finestra.
- Le varie statistiche possono poi essere pesate in modo diverso per tenere conto della correlazione delle risposte.

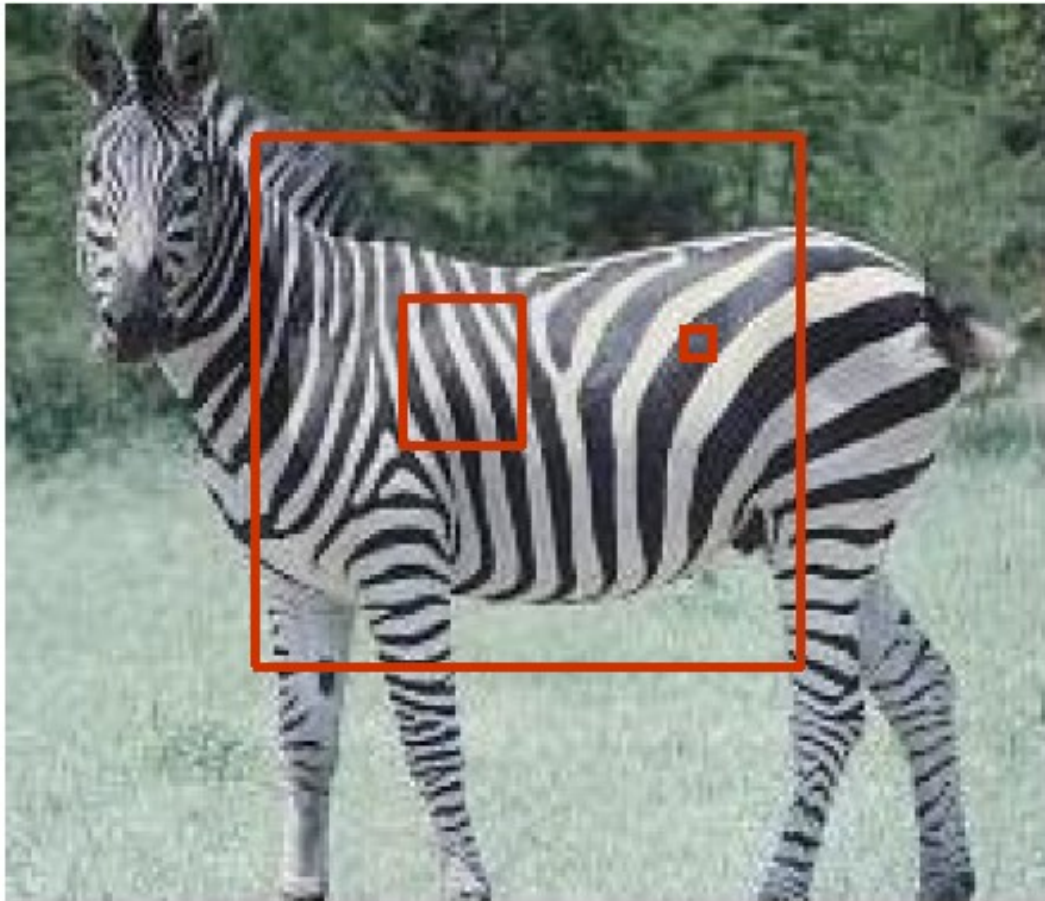
Rappresentazione di texture



Texture di query

Risultato della ricerca in un database di textures

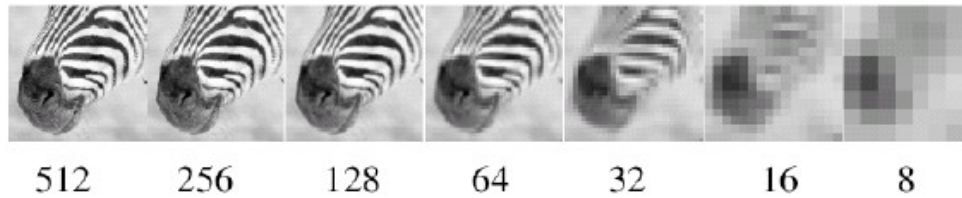
Scelta della scala



- Immaginiamo di prendere un punto dell'immagine e partire con una finestra piccola attorno al punto. Facciamo poi crescere progressivamente la finestra finché non ci sono cambiamenti significativi.
- Possiamo tracciare i cambiamenti della polarità dell'immagine, ovvero, calcoliamo la direzione del gradiente dominante nella finestre e facciamo le medie delle proiezioni positive e negative dei gradienti dei vari pixels su quello dominante. La differenza delle due medie ci fornisce un'indicazione della polarità.
- Il risultato dipende dalla risoluzione dell'immagine e il livello di scala da cui partiamo. Se la risoluzione è alta tale da vedere un pelo del manto, potremmo fermarci ad una texture di peli. Partendo da una scala un po' più alta ci fermiamo ad una texture di striature.

Analisi di immagini

- Il processo di convoluzione di un'immagine con un banco di filtri è detto di **analisi**.
- Un esempio di analisi di immagine mediante filtri di smoothing è la piramide Gaussiana.



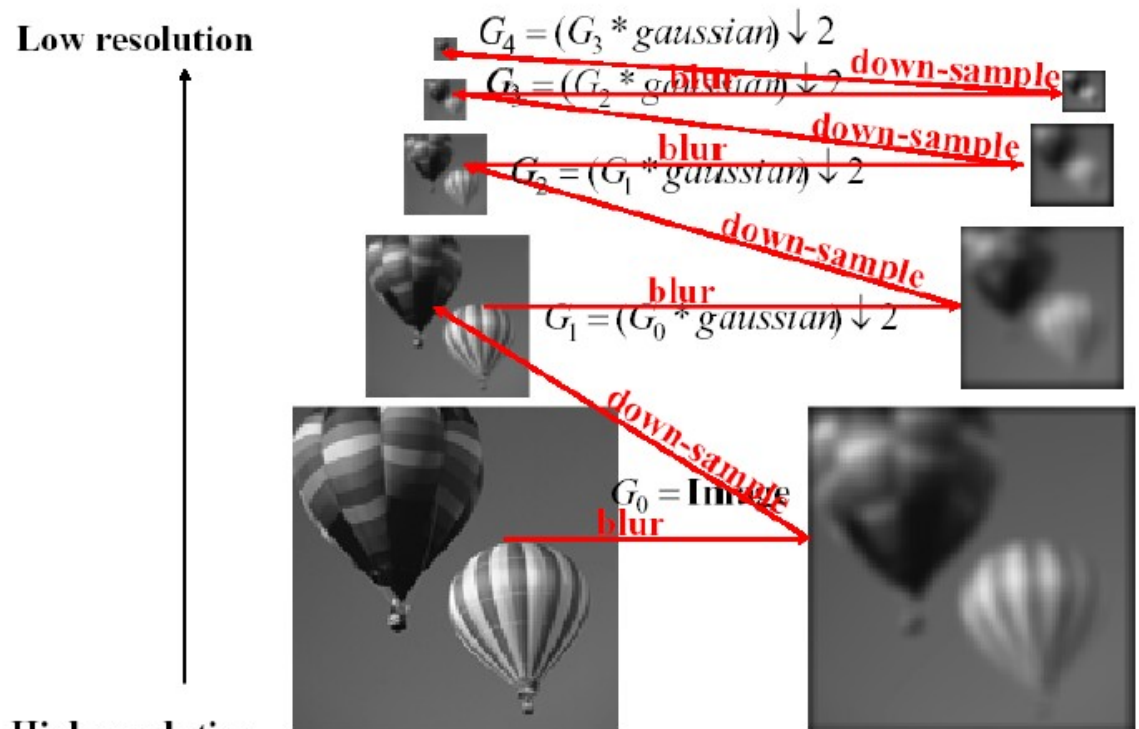
Piramidi di immagini

- Una piramide di un'immagine è una collezione di rappresentazioni di quell'immagine a scale diverse.
- Tipicamente ogni strato della piramide ha una lunghezza che è la metà dello strato precedente e il doppio del successivo e il parametro della Gaussiana è scelto tra 1 e 2.
- Nel caso di una piramide Gaussiana, ad ogni strato viene applicato un filtro Gaussiano e il risultato viene poi sottocampionato per ottenere lo strato successivo.
- Il sottocampionamento nel caso di fattori di scala $\frac{1}{2}$ può essere fatto prendendo i pixels con entrambe coordinate pari.

$$P_G(I)_n = \begin{cases} I & n = 1 \\ S^\downarrow(G_\sigma * P_G(I)_{n-1}) & 1 < n \leq N \end{cases}$$

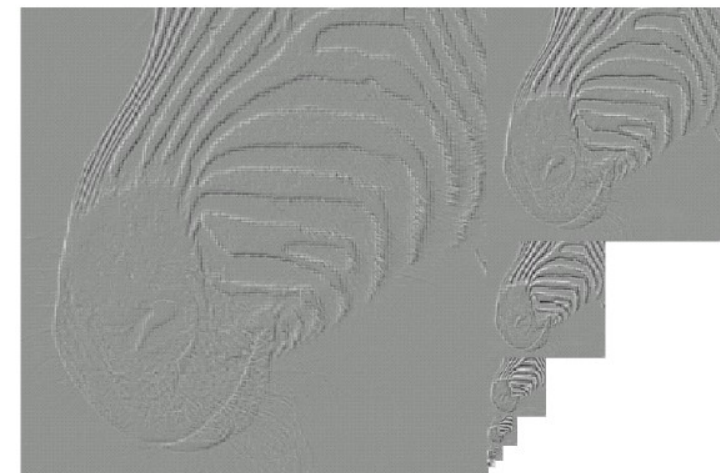
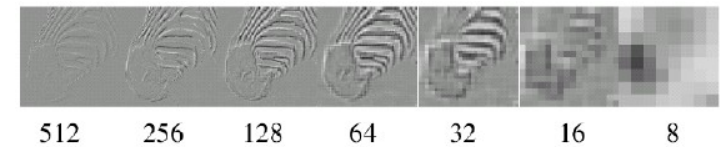
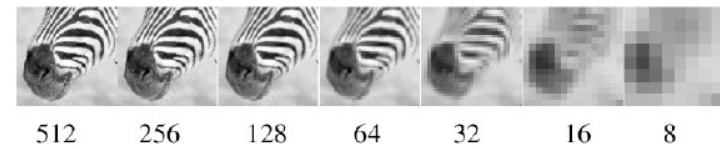
operatore di sottocampionamento

altezza piramide



Piramidi di immagini

- Una piramide Gaussiana è molto ridondante perché intuitivamente mantiene sempre le informazioni alle basse frequenze e rimuove via via sempre più informazione alle alte frequenze.
- Ogni strato della piramide Gaussiana, se supercampionato in modo opportuno, diventa una predizione approssimata dello strato precedente e la differenza tra lo strato effettivo e quello predetto produce un'immagine di errore.
- Prendendo le immagini di errore per ogni strato otteniamo una piramide detta Laplaciana.
- Il sovracampionamento, nel caso di fattori di scala $\frac{1}{2}$, trasforma ogni pixel in un quadrato 2×2 .



operatore di
sovracampionamento

$$P_L(I)_n = \begin{cases} P_G(I)_N & n = N \\ P_G(I)_n - S^\uparrow(P_G(I)_{n+1}) & 1 \leq n < N \end{cases}$$

Sintesi di immagini

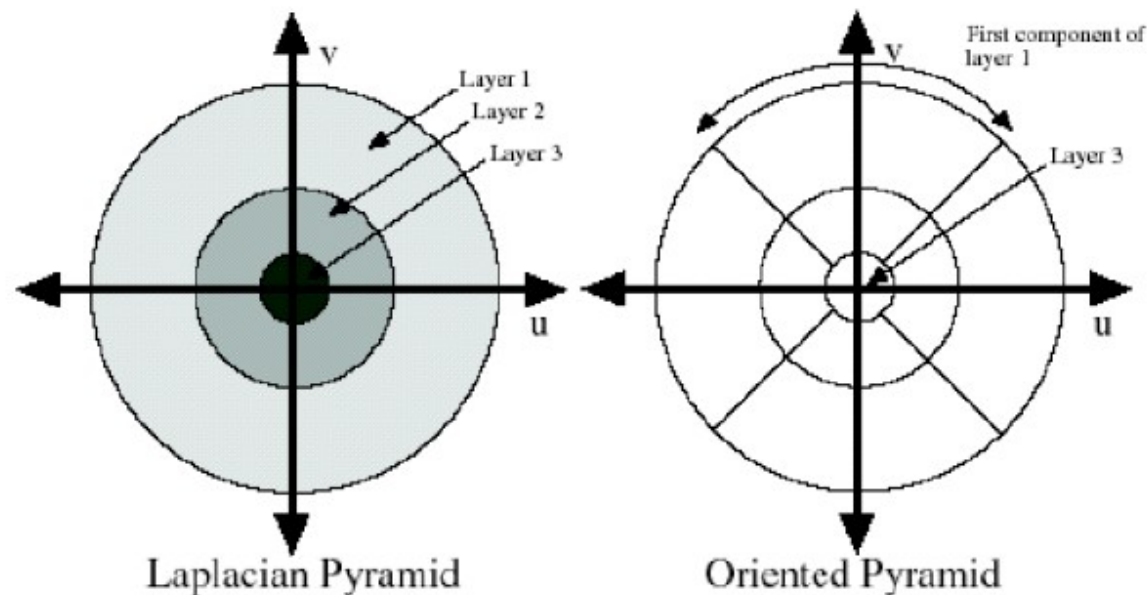
- Il processo di sintesi ricostruisce un'immagine a partire dalla piramide Laplaciana.
- In particolare, dalla piramide Laplaciana possiamo ricostruire in senso inverso l'intera piramide Gaussiana, il cui strato ultimo è l'immagine originale.

$$P_G(I)_n = \begin{cases} P_L(I)_N & n = N \\ S^\uparrow P_L(I)_{n+1} + P_L(I)_n & 1 \leq n < N \end{cases}$$

$$I = P_G(I)_1$$

Piramidi Laplaciane vs Orientate

- Ogni piramide Laplaciana può essere vista come il risultato di un filtro passa banda in cui il primo strato contiene frequenze alte e l'ultimo frequenze basse ed ogni strato intermedio rispecchia una certa banda di frequenze.
- In pratica, ogni strato è una sezione ad anello con centro nell'origine del dominio delle frequenze.
- La piramide Laplaciana non porta informazioni di orientamento.
- Una piramide orientata aggiunge rispetto a quella Laplaciana una sezione degli anelli per bande in modo da discriminare anche l'orientamento e non solo la frequenza.



Filtri di Gabor

- Uno svantaggio legato all'analisi di Fourier è la perdita dell'informazione spaziale, in quanto ogni valore nel dominio delle frequenze è determinato da tutti i pixels nel dominio spaziale.
- Per ovviare a questo possiamo ricorrere a filtri di Gabor.
- I filtri di Gabor sono filtri simili alle immagini base di Fourier, ma sono applicati ad una porzione locale dell'immagine per mantenere la maggior parte dell'informazione spaziale.

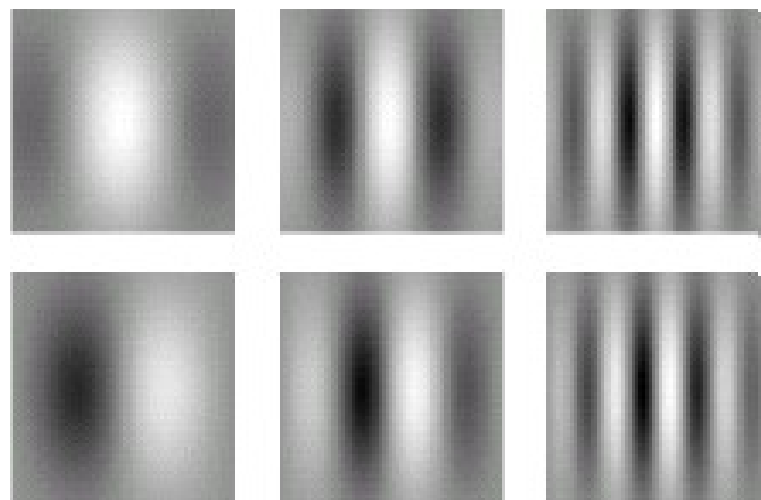
Immagine base

Finestra

$$G[x, y] = e^{i2\pi(k_0x + k_1y)} e^{-(x^2 + y^2)/2\sigma^2}$$

$$G_{sim}[x, y] = \Re(G[x, y])$$

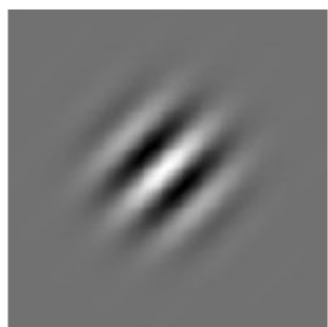
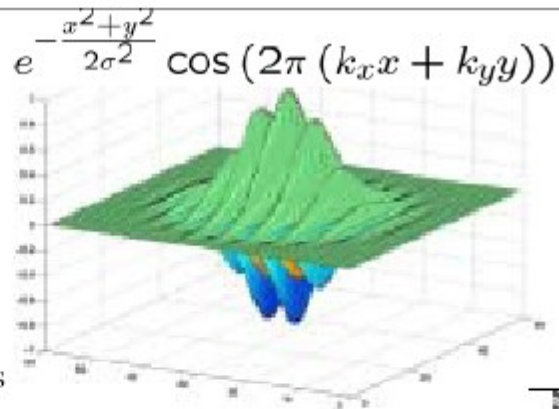
$$G_{antisim}[x, y] = \Im(G[x, y])$$



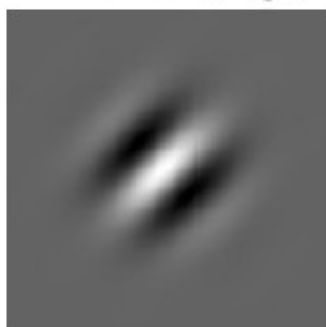
Filtri di Gabor



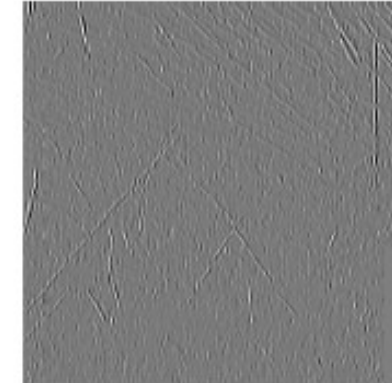
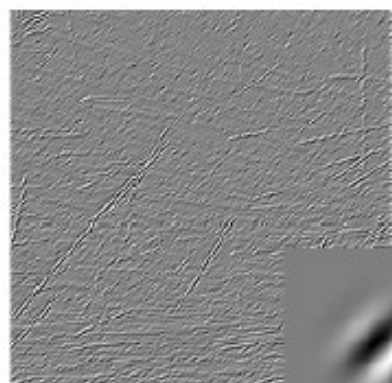
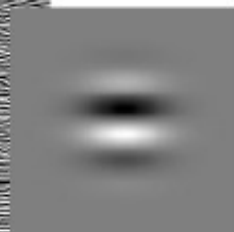
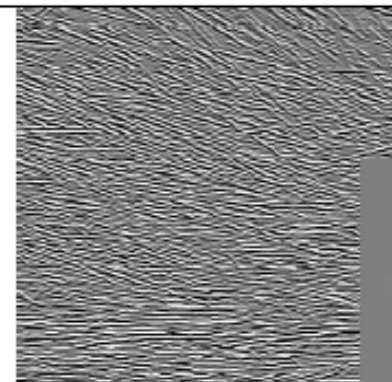
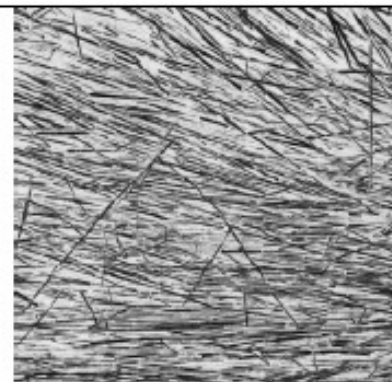
High frequency along axis



Lower frequency

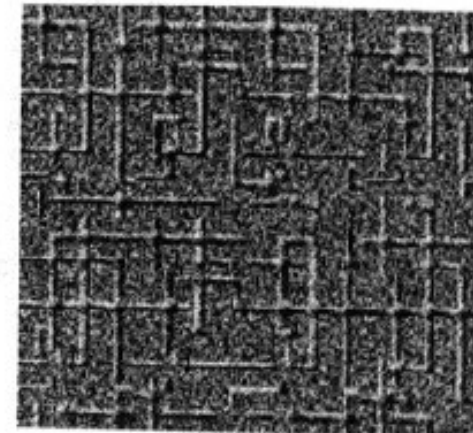
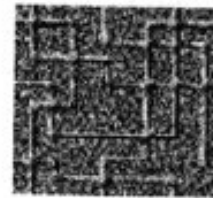
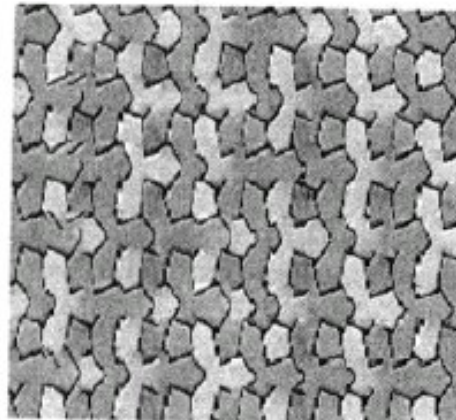
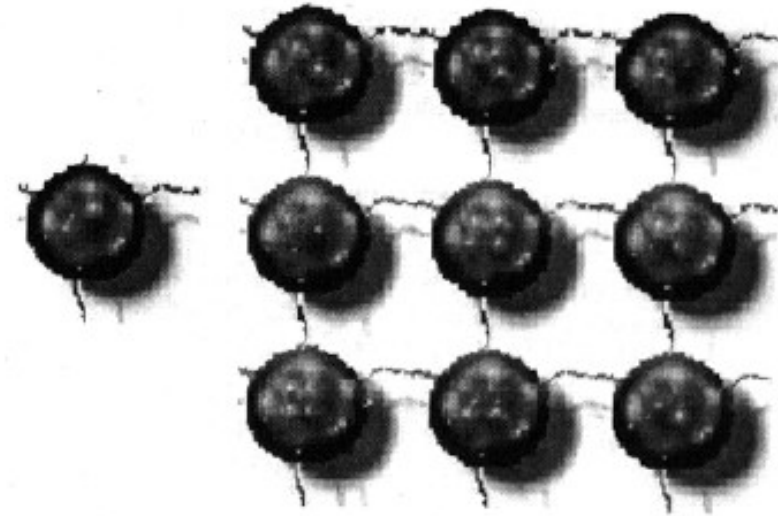
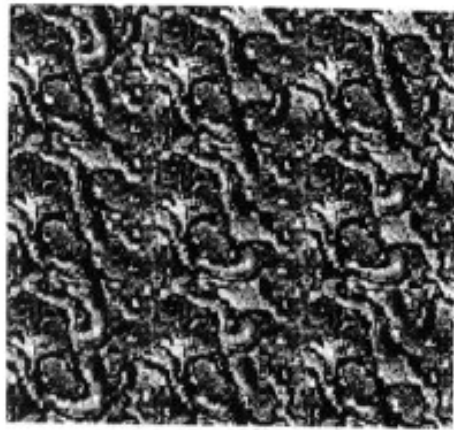


Even lower frequency



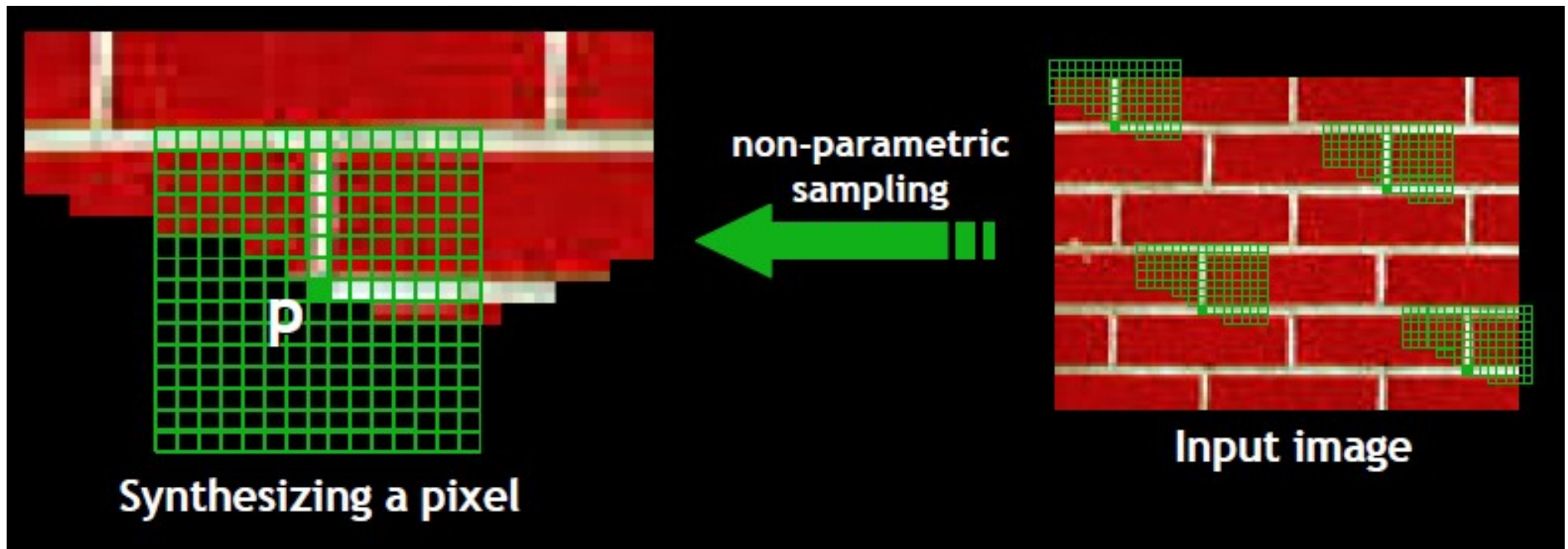
Sintesi di texture

- La **sintesi di texture** cerca di creare nuove superfici o riempire parti mancanti, partendo da una texture di esempio.

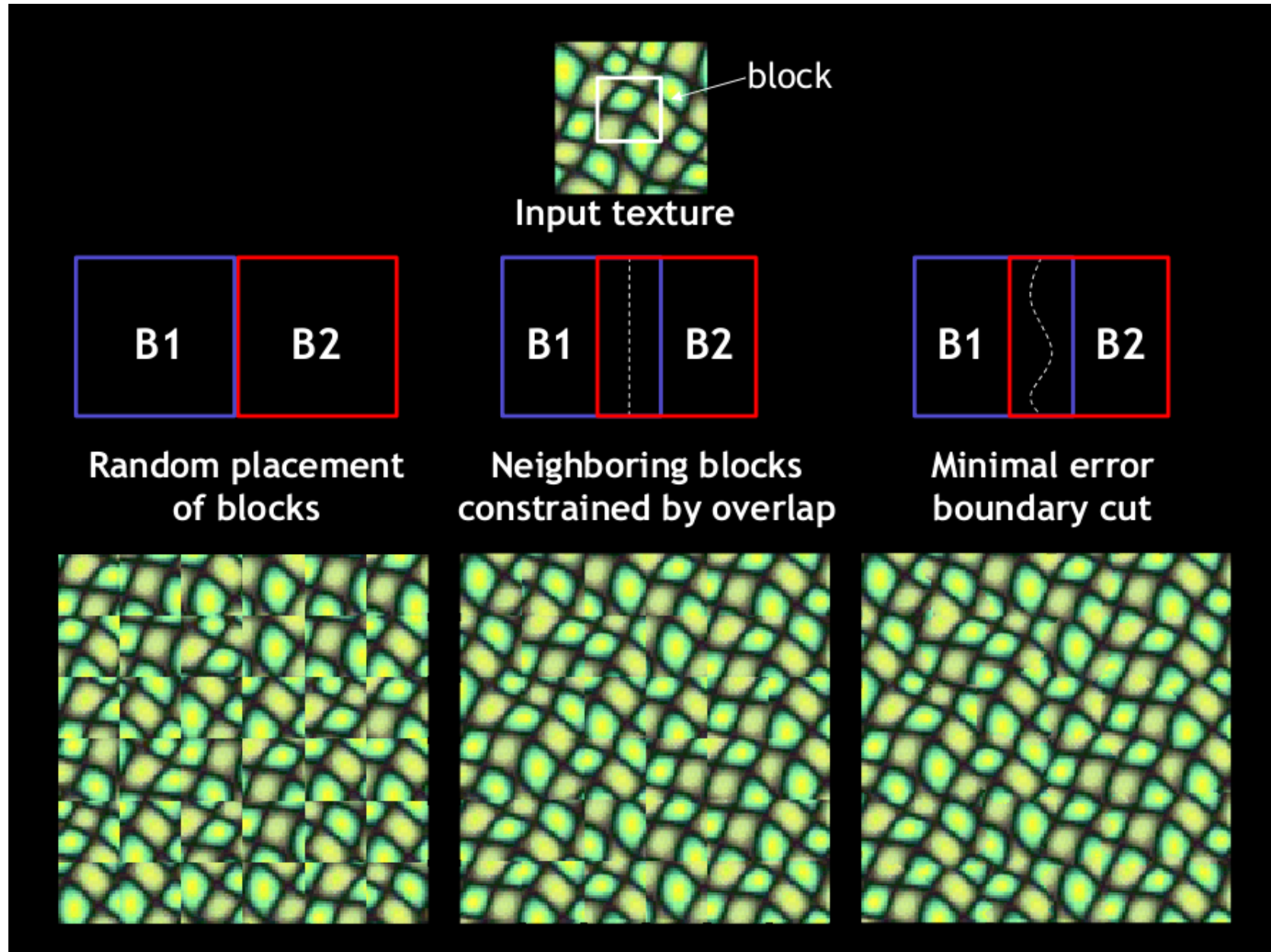


Algoritmo di Efros & Leung (1999)

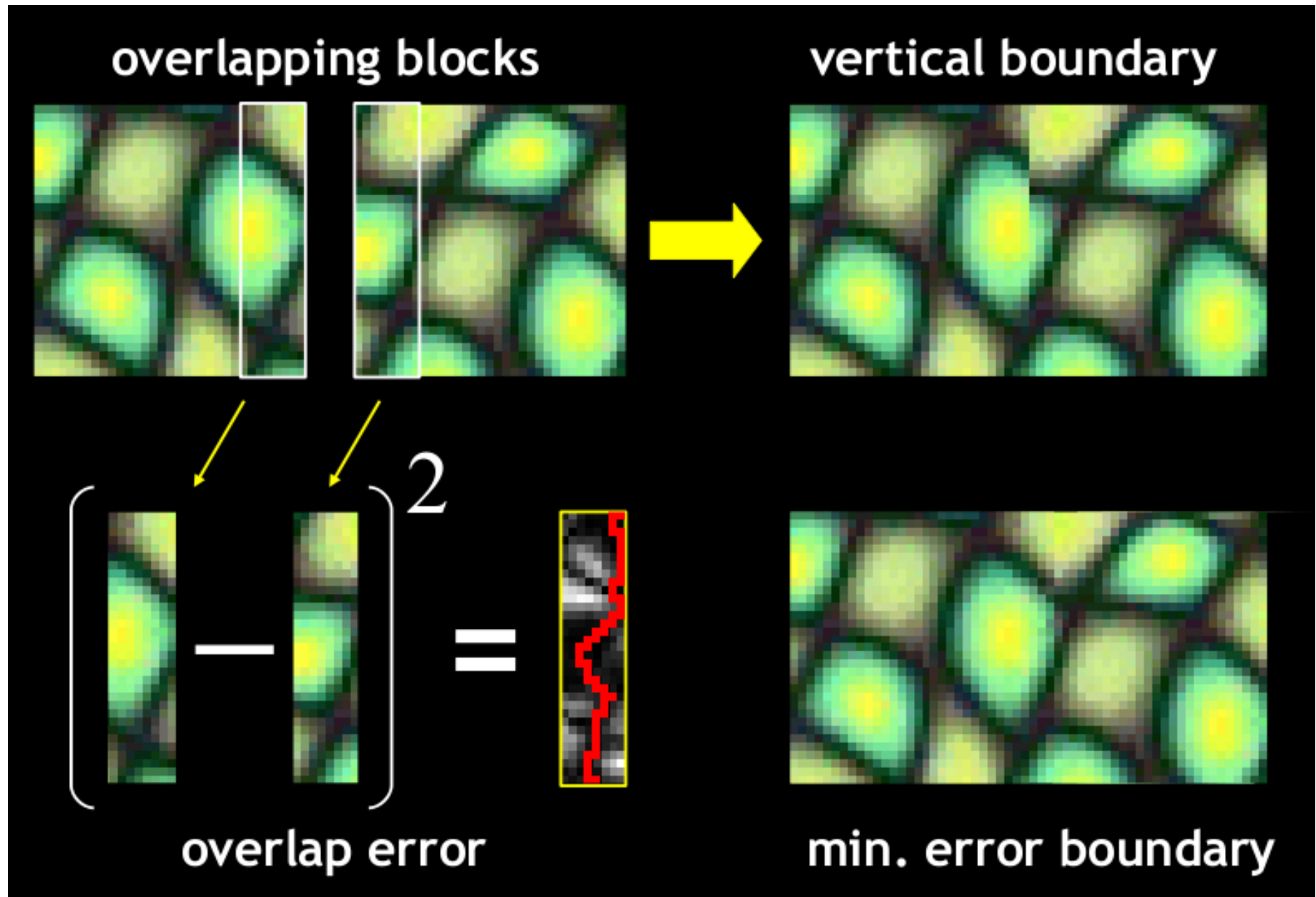
- Usare l'immagine data come sorgente di un modello probabilistico.
- Scegliere il valore dei pixels non-sintetizzati comparando i valori sintetizzati in un intorno del pixel con la texture di esempio. Abbiamo un match se la somma (pesata con gaussiana centrata sul pixel) di differenze al quadrato è inferiore ad una certa soglia.
- Si sintetizzano prima i pixels con un'intorno più ricco (con più vicini sintetizzati). I pixel che non trovano corrispondenze vengono rielaborati alla fine.



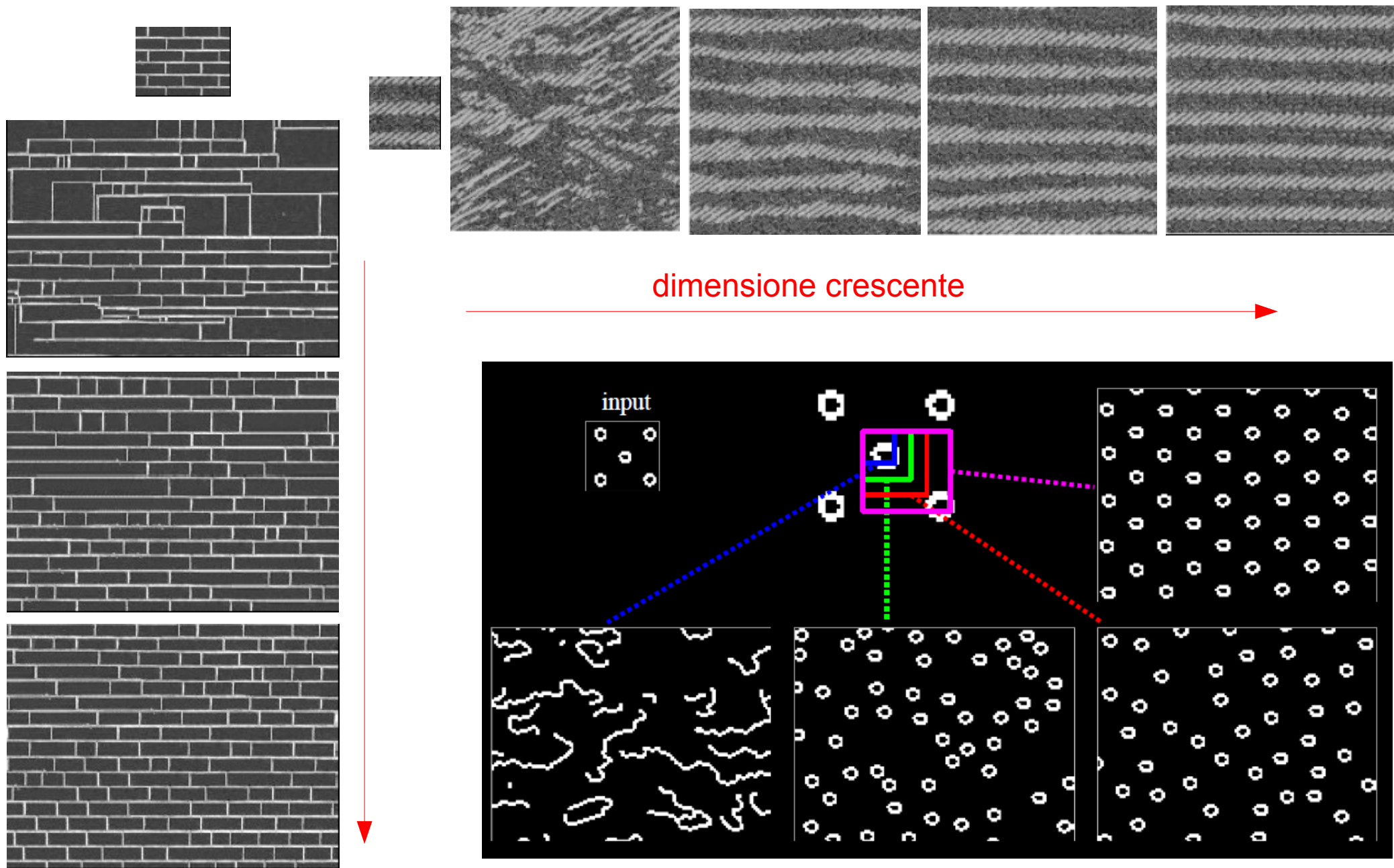
Per velocizzare [Efros Freeman]



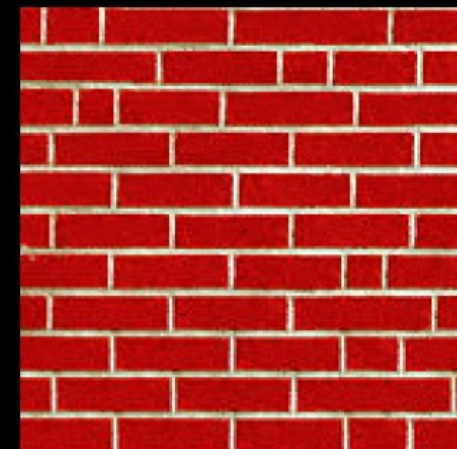
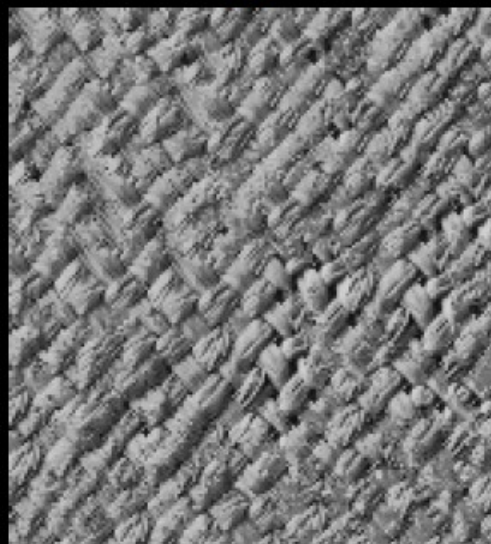
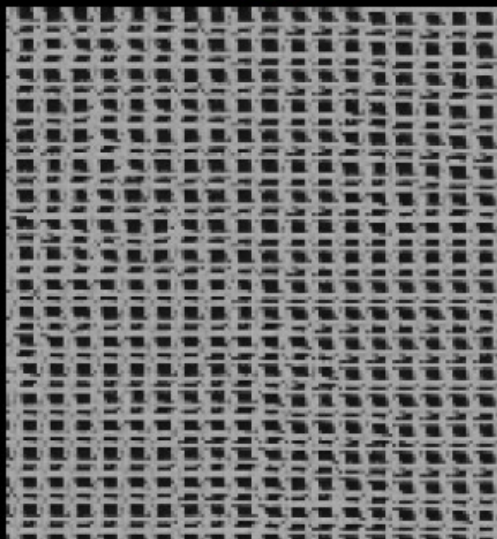
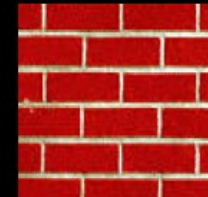
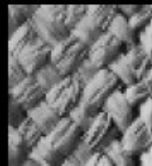
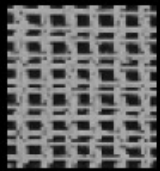
Per velocizzare ...



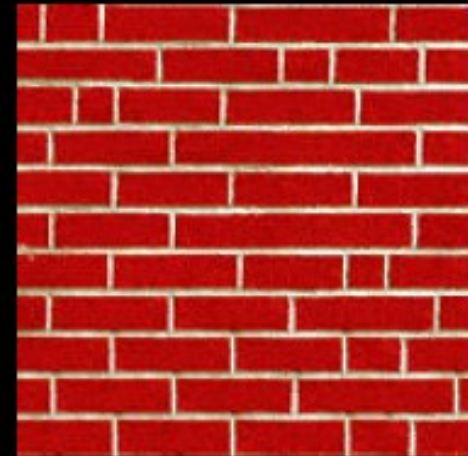
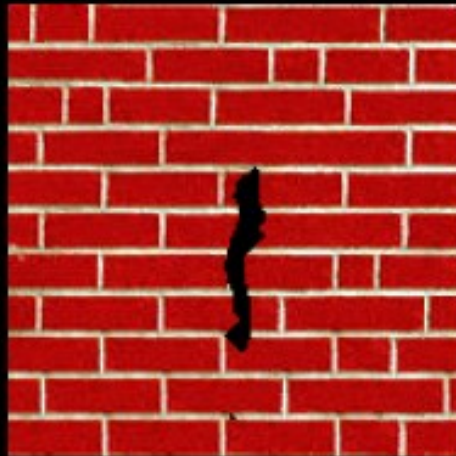
Dimensione della finestra



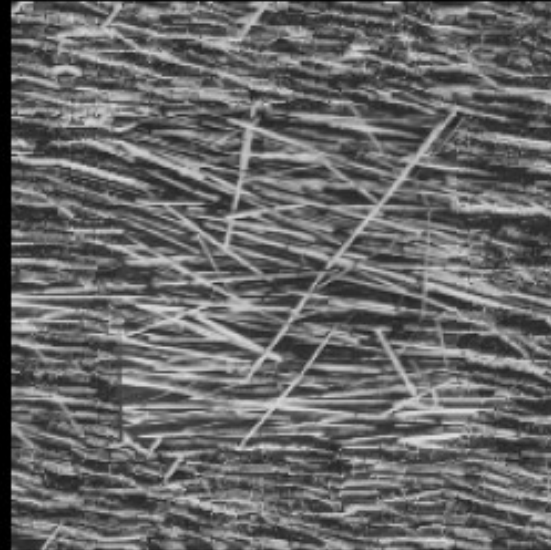
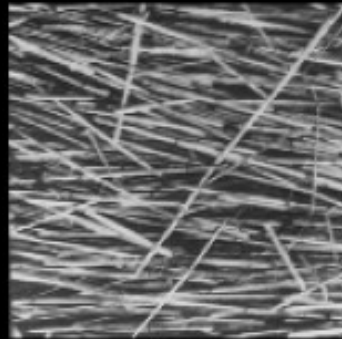
Sintesi di texture



Sintetizzazione di parti mancanti

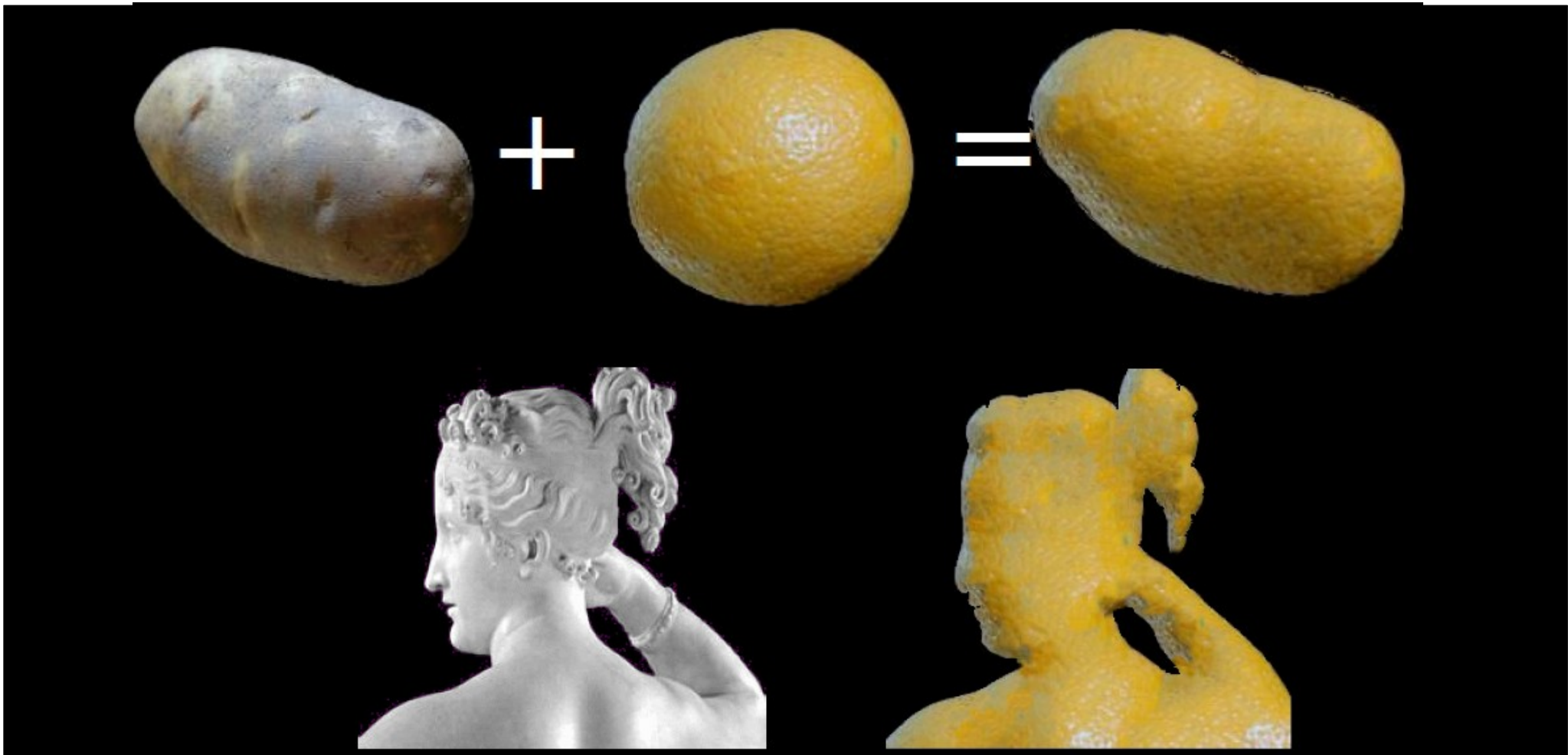


Estrapolazione



Trasferimento di Texture

- Il trasferimento di texture è un'interessante applicazione in cui si vuole rappresentare un'immagine utilizzando parti provenienti da una texture.
- L'algoritmo è uguale a quello di sintesi con l'aggiunta del vincolo che ci deve essere una similarità tra parti dell'immagine e parti di texture sintetizzate.



Trasferimento di Texture

parmesan



+



=



rice



+



=

