

Capitolo 2

Analisi Automatica di *Pattern/Texture* Visuali

1. Concetti preliminari

Diversi tentativi per caratterizzare in termini esatti e formali il concetto primitivo di *pattern*, di tessitura o *texture* sono stati fatti in passato e come tali sono presenti in letteratura; tutti comunque presentano dei limiti e sono spesso applicabili solo in contesti specifici. In generale è però possibile osservare come un gran numero di fenomeni naturali presentino una qualche regolarità di natura stocastica nonostante lo stesso fenomeno risulti essere per sua natura non predicibile o addirittura sconosciuto. Esempi tipici di *texture* sono ad esempio

l'immagine di una scacchiera o la pelle di un leopardo. Risulta chiaro come comunque il concetto di omogeneità o di regolarità non è oggettivamente definito in quanto ha a che fare con le peculiarità percettive di ogni singolo individuo. Si veda (Bergen, Julesz - 1993) per una attenta ed oculata panoramica al riguardo. Una regione in un'immagine è detta *textured* se presenta caratteristiche di omogeneità relativamente alle sue caratteristiche percettive (intensità luminosa, saturazione, regolarità strutturale ecc.). Ovviamente lo studio e l'analisi di tessiture in *Computer Vision* diventa tanto più problematico quanto più complessa è la scena in cui essa viene a trovarsi. Le tessiture reali inoltre risultano essere per la maggior parte costituite da elementi strutturali di base, detti *texton* (Vistnes - 1989), la cui regolarità è per lo più di natura stocastica. Da questo punto di vista le tessiture, rappresentano campioni di distribuzioni di probabilità e vengono analizzate con i mezzi propri della statistica. Si tende cioè a caratterizzare in maniera più o meno esplicita la legge statistica, o il che è lo stesso, il modello di probabilità da cui l'immagine stessa è stata generata. Riuscire in questo intento è spesso difficile in situazioni reali, soprattutto quando la conoscenza del processo generativo di base, è parzialmente o del tutto sconosciuto. Nasce quindi l'esigenza di utilizzare tecniche capaci di apprendere le caratteristiche peculiari del *pattern* e di riprodurre esemplari simili almeno dal punto di vista percettivo. Esistono inoltre chiare evidenze sperimentali nel campo delle scienze cognitive che portano inevitabilmente a considerare l'analisi di tessiture come una delle specializzazioni del sistema visivo umano. La ricerca di base sulle tessiture è stata avviata da Julesz che nel 1962 pubblicò un articolo in cui riuscì a fare il punto, ponendo il problema in termini meramente scientifici ma soprattutto riuscendo a focalizzare,

attraverso due semplici quesiti, quelli che a tutt'oggi sono i principali obiettivi che ci si propone di raggiungere nello studio delle tessiture. Il primo riguarda l'individuazione delle caratteristiche peculiari o statistiche che individuano univocamente un *pattern* grafico, una *texture*, tali cioè da caratterizzarne l'indiscriminabilità percettiva, mentre il secondo, più specificatamente rivolto all'ambiente matematico/statistico, si chiede come riuscire, una volta individuato un insieme di caratteristiche o *feature* di tipo statistico, a generarne campioni pseudo-casuali che ne seguano e/o ne condividano, riproducendone l'andamento, tali proprietà di base.

2. *Texture e Computer Vision*

In *Computer Vision*, la ricerca di base analizza le tessiture cercando di caratterizzarne la regolarità intrinseca catturandone così in maniera esplicita le specifiche peculiarità. I principali obiettivi e compiti che in tale ambito ci si prefigge sono:

- La ***Discriminazione*** o ***Detection*** di una tessitura in un'immagine che tende a separare, trovandone i bordi, le zone con tessiture differenti. Questa operazione può risultare particolarmente complicata in presenza di informazioni legate solamente all'intensità dei *pixel* dell'immagine;
- L'***Analisi*** e ***Sintesi*** di tessiture che si propone invece di analizzare la natura stocastica della tessitura stessa, con lo scopo di riuscirne a catturare, in maniera più o meno diretta, il processo generativo intrinseco che la caratterizza; in questo modo si riescono a riprodurre, nella fase di sintesi,

texture che percettivamente appaiono differenti dall'originale pur appartenendo alla stessa classe statistica di base. Si veda ad esempio la Fig. 1 dove sono riportati a titolo esemplificativo alcuni esempi di sintesi di tessiture. Nelle sezioni successive verranno illustrate nei dettagli alcune delle tecniche più significative presenti in letteratura mentre nel capitolo successivo verranno presentati in dettaglio i risultati ottenuti in quest'ambito attraverso l'implementazione di tecniche e di algoritmi *ad-hoc* facenti uso di procedure di clusterizzazione e di un approccio multirisoluzione;

- La **Classificazione** che tende appunto a classificare le tessiture ed in questo senso è quindi strettamente legato alla operazione di analisi. Classificare una tessitura è molto più difficile che discriminarla. Infatti la stessa tessitura può apparire in un'immagine in modi anche assai diversi tra loro, e ciò dipende ovviamente dal grado di luminosità, dall'angolazione, dalla scala di risoluzione, ecc. In un certo senso bisogna quindi riuscire a comprendere la semantica della tessitura stessa per poterla ben classificare. In quest'ambito è possibile individuare applicazioni che riguardano ad esempio il riconoscimento di immagini satellitari SAR (*Synthetic Automatic Radar*).

3. Analisi e Sintesi di Tessiture

La generazione automatica di tessiture, detta tecnicamente *sintesi*, è sicuramente uno dei campi di ricerca più attivo degli ultimi anni. In questa sezione richiamiamo brevemente, seguendone il giusto iter temporale, le principali tecniche ed algoritmi presenti in letteratura.

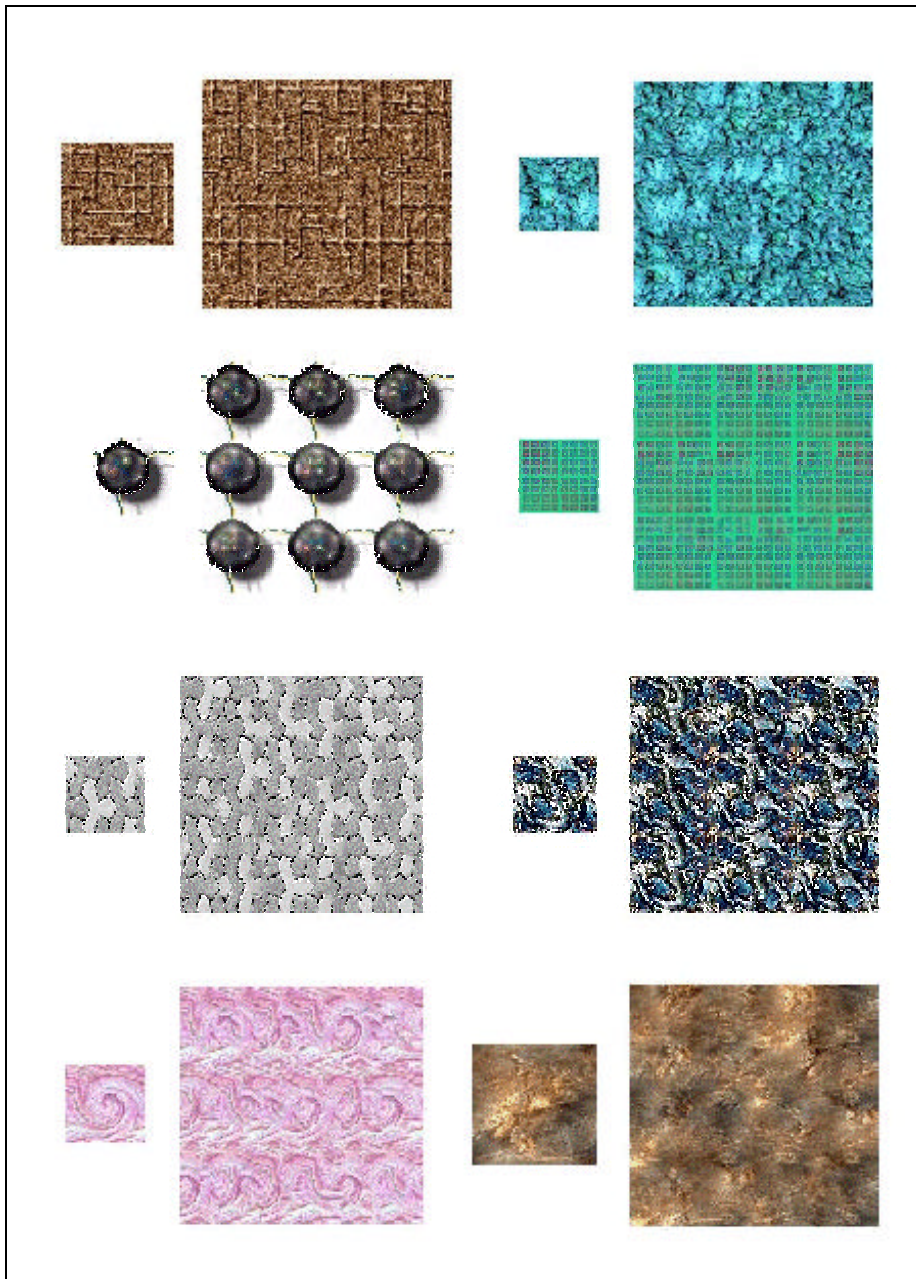


Fig.1. Sintesi di Texture

L'approccio di natura probabilistica/statistica di cui discuteremo ampiamente nel seguito della Tesi è relativamente recente; i primi lavori in questo campo non consideravano affatto le tessiture come campioni estratti da distribuzioni di probabilità. L'approccio più comunemente utilizzato era quello di cercare di

riprodurre artificialmente il processo fisico, che stava dietro alla tessitura sotto osservazione.

L'obiettivo ideale in questo campo, così come enunciato dallo stesso Julesz, sarebbe quello di riuscire a trovare una sorta di teoria del *tristimolo* che, come noto, afferma che un qualsiasi colore visibile può essere univocamente determinato mediante una combinazione lineare di tre colori di base. Nella teoria delle tessiture ciò equivarrebbe a trovare le giuste *feature* statistiche di base da combinare opportunamente tra di loro secondo una data modellizzazione matematica.

Sempre Julesz fu il primo a proporre come soluzione lo studio delle cosiddette statistiche associate alle co-occorrenze del k -esimo ordine. In particolare egli congetturò che per studiare completamente una qualsiasi tessitura fosse sufficiente conoscerne i momenti di ordine 2. Successivamente però si scoprì (Diaconis, Freedman - 1981) che esistono comunque tessiture con identiche statistiche del secondo e del terzo ordine che risultano essere percettivamente distinguibili. Un'altra tecnica di base, presente in letteratura consiste nel considerare le tessiture, come oggetti complessi ottenuti combinando tra loro gli elementi strutturali di base della tessitura stessa, detti *texton* (Vistnes - 1989) mediante opportune primitive di basso livello.

Il percorso scientifico volto alla individuazione delle *feature* statistiche più significative nell'analisi di *texture* non si è comunque fermato alle co-occorrenze del k -esimo ordine, sia per quanto riguarda la modellizzazione che la discriminazione. Citiamo per esempio gli approcci legati alle matrici di co-occorrenza (Haralick - 1979), alle statistiche di tipo *run-length* (Weszka, Dyer,

Rosenfeld - 1976), i già citati *texton* insieme a dozzine di altre misure specifiche. La maggior parte di questi approcci presentano però una limitata potenza espressiva, e come tali risultano applicabili solo in contesti molto limitati.

Il vero di salto di qualità è stato compiuto nei primi anni '80 con l'introduzione in letteratura di potenti ed espressivi strumenti matematici quali i filtri di Gabor (Daugman - 1985), i filtri a piramide (Simoncelli, Freeman, Adelson, Heeger - 1992) e le trasformate di tipo *wavelets* (Daubechies - 1992). Un ulteriore passo in avanti è stato fatto sostituendo le statistiche del k -esimo ordine, con lo studio di apparentemente semplici statistiche di base calcolate sugli istogrammi, sia marginali che congiunti (Heeger, Bergen - 1995; De Bonet - 1997; De Bonet, Viola - 1998). Tali metodi rappresentano a tutt'oggi quelli che riescono a tirare fuori i risultati più significativi ed *impressive*, soprattutto dal punto di vista percettivo, si veda ad esempio il sito Web (De Bonet - 1999) per un'esaustiva *demo* al riguardo. D'altra parte questi metodi, altamente significativi dal punto di vista ingegneristico, non sono in grado di essere generali al punto da modellizzare il fenomeno in maniera rigorosa.

Dal punto di vista più strettamente matematico il primo algoritmo, per così dire stocastico, per la modellizzazione di tessiture fu proposto da Cross e Jain nel 1983, che introdussero un MRF auto-binomiale definito su coppie opportune di *clique*.

Obiettivo di questo capitolo è quello di riuscire ad illustrare pregi e difetti di due delle principali tecniche presenti in letteratura quella che fa capo al filone Zhu, Wu, Mumford relativamente al modello probabilistico FRAME basato sull'apporto combinato dei *Markov Random Field* e della teoria dell'entropia

minimax, e l'approccio multirisoluzione che iniziato da Heeger e Bergen nel 1995 ha trovato in De Bonet successivamente la definitiva consacrazione. Tali lavori infatti rappresentano la base scientifica e il *background* culturale, che è alla base dei risultati presenti nel capitolo successivo.

Nel paragrafo finale poi viene brevemente illustrato l'approccio *Hypertexture* di Perlin (1989), volto alla rappresentazione realistica in ambienti 3D di tessiture, ottenuta grazie all'utilizzo combinato di funzioni di base legate alla forma e ad alcune primitive di massima legate alla cosiddetta apparenza percettiva della tessitura stessa quali inclinazione, turbolenza, effetto rumore, ecc.

4. Il Modello FRAME e l'Insieme di Julesz

Abbiamo già definito nel capitolo precedente il concetto di MRF, *Markov Random Field*, descrivendone sia pur per grandi linee le idee di base ed alcune delle applicazioni specifiche presenti nell'ambito dell'elaborazione delle immagini. Nel 1983 Cross e Jain pubblicarono un articolo, in cui tali concetti vennero applicati per la prima volta per modellare e sintetizzare tessiture. Una generica tessitura, viene ad essere trattata come uno *stochastic image field*, in cui cioè la natura prettamente casuale della stessa viene modellata attraverso un modello binomiale. Si suppone cioè che ciascun punto della tessitura sia governato da una distribuzione binomiale i cui parametri sono controllati dai rispettivi vicini, in funzione anche del numero di toni di grigio dell'immagine stessa.

Il metodo in oggetto è in grado di ottenere campioni specifici visivamente apprezzabili, soprattutto per quanto riguarda le cosiddette *microtextures*, piccole

parti della tessitura originaria che ne caratterizzano i dettagli. Ovviamente cruciale risulta essere la scelta dei parametri del modello. Si rimanda all'originale articolo per ulteriori dettagli (Cross, Jain - 1983). Nonostante la classe di tessiture, effettivamente modellata da tale approccio sia esigua, Cross e Jain ebbero il merito specifico di fare da battistrada ad una serie sterminata di lavori specifici, che hanno cercato, nel corso degli anni, con alterne fortune, di ottenere un modello esplicito, computazionalmente effettivo, il più generale possibile, in grado di rendere trattabile il problema anche per tessiture, reali o meno, altamente *complesse*.

L'utilizzo dei MRF, quindi rimane uno degli strumenti più adeguati, da un punto di vista matematico, per la modellizzazione statistica di immagini digitali. La difficoltà specifica nell'utilizzo di tali modelli è, come ovvio, il *design* della funzione energia che deve soddisfare le proprietà chiave o di base del fenomeno visuale sotto osservazione. Recentemente sono stati introdotti in letteratura due nuove classi di modelli di tipo MRF, una che modella vincoli geometrici attraverso l'utilizzo intensivo della morfologia matematica (Sivakumar, Goutsias - 1999) tuttora in fase di ulteriore sviluppo, e l'altra, già introdotta nel capitolo precedente, cui fanno riferimento i modelli *FRAME*, dovuta a (Zu, Wu, Mumford - 1998).

I modelli di tipo *FRAME*, nonostante siano in qualche modo derivati è comunque abbiano uno stretto legame con i *Markov Random Field* per così dire classici, presentano una differente filosofia di base. Essi infatti, tendono a favorire l'apprendimento e la specifica di interazioni locali introducendo delle funzioni non lineari legate alla risposta ad opportuni filtri, senza essere quindi fortemente

vincolati dal cosiddetto *neighborhood size*, che nel caso classico rende computazionalmente intrattabile il problema.

Anche se i risultati conseguiti, si veda ad esempio la Fig 2, sono significativi, risulta comunque necessario, sempre per motivi strettamente computazionali, riuscire a scegliere comunque un insieme di filtri opportunamente limitato.

In questo paragrafo ci limiteremo a presentare brevemente, gli ulteriori sviluppi nonché le possibili estensioni degli studi riguardanti tale approccio (Zu, Liu, Wu - 1999).

Dato un insieme di misure statistiche e un insieme di immagini di tipo *texture* aventi uno specifico *pattern*, l'*insieme di Julesz* è definito come l'insieme di tutte le possibili immagini (possibilmente infinito) che ne condividono tale comportamento statistico. Da un punto di vista teorico allora, il problema della sintesi di una tessitura, individuato l'appropriato insieme di *feature* statistiche, si ridurrebbe ad una semplice estrazione o campionamento, da eseguire su tale insieme. E' possibile quindi associare il relativo *insieme di Julesz*, a ciascun *texture/pattern* a condizione di averne individuato il giusto insieme di *feature* statistiche che lo descrivono o, il che è lo stesso, dato un insieme di immagini che si suppone essere estratto da tale insieme individuarne tali *feature*.

Si dimostra che l'insieme di Julesz, così come definito è consistente con la classe di tessiture individuato da un modello *FRAME*. E' allora possibile, individuata una strategia per rendere più efficiente la costruzione di tale modello, utilizzare tale metodologia per costruire modelli di tessiture consistenti ed altamente espressivi.

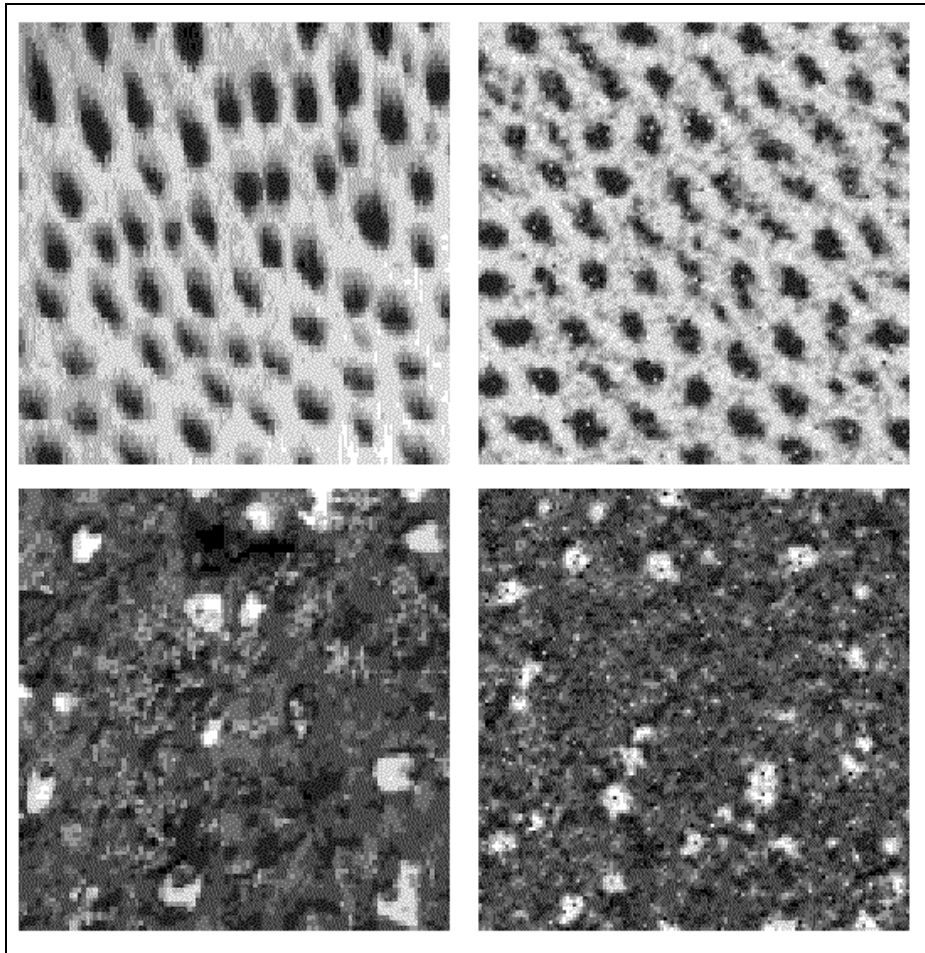


Fig. 2. Risultati sperimentali ottenuti utilizzando il modello FRAME nella forma base.

I due rilevanti *improvement* presenti in (Zu, Liu, Wu - 1999) riguardano:

- Un nuovo ed efficiente algoritmo di campionamento detto *Window Gibbs Sampler*: il classico *Gibbs sampling*, di cui abbiamo ampiamente discusso nel capitolo precedente, viene adattato al problema in oggetto. Anziché andare a rimpiazzare di volta in volta un singolo *pixel*, si valuta opportunamente la possibilità di rimpiazzare un'intera maschera di valori, facendo comunque uso della relativa probabilità condizionale. Si noti infatti

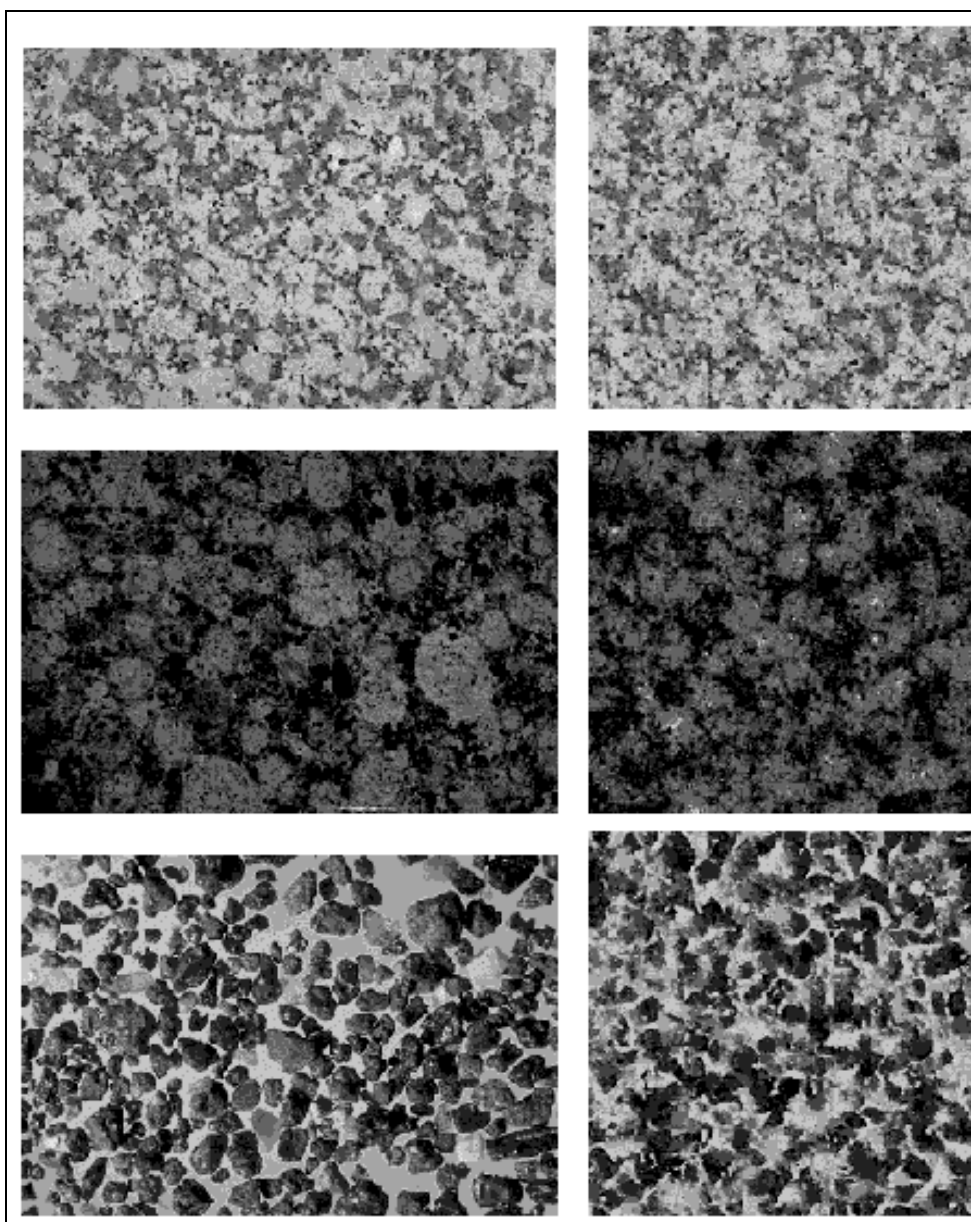


Fig.3. Esempi di sintesi di tessiture utilizzando il modello *FRAME* esteso, ottenuti utilizzando 56 filtri.

- come nel modello *FRAME* le singole *feature* sono proprio legate alla risposta ad opportuni filtri e quindi alle relative maschere di convoluzione;
- Una opportuna riduzione dello spazio delle *feature* utilizzate, legate soprattutto al numero di *bins* da utilizzare nello studio degli istogrammi

delle singole *feature* in oggetto. Le *feature* statistiche che sembrano essere allo stato attuale le più "espressive" sono proprio gli istogrammi della risposta ad opportuni filtri, a diversi livelli di risoluzione, i cosiddetti *istogrammi marginali congiunti*. Si tratta cioè di istogrammi multidimensionali, che introducono fattori di complessità elevati e che vanno quindi gestiti opportunamente.

La Figura 3 mostra i dettagli di alcuni dei risultati ottenuti, facendo uso di tale approccio.

5. Texture e Multirisoluzione (1): Heeger e Bergen

Il lavoro di Heeger e Bergen del 1995, di cui tratteremo in questo paragrafo, riveste un ruolo di primo piano nel settore specifico dell'analisi/sintesi di tessiture. Tale importanza risiede nel fatto che in tale articolo, per la prima volta, si fa uso esplicito di un'analisi multirisoluzione del *pattern* oggetto di studio. Adducendo infatti motivazioni di carattere psicofisico, sulla percezione umana, gli autori propongono infatti un algoritmo altamente efficiente ma soprattutto in grado di riprodurre un discreto ed espressivo insieme di *texture*.

Vengono utilizzate due tipi di rappresentazione a piramide dell'immagine: la piramide *Laplaciana* (Burt, Adelson - 1983) e quella *Steerable* (Simoncelli, Adelson - 1990). Come noto, la prima rappresenta un'immagine ai diversi livelli di risoluzione catturandone a ciascun livello informazioni dettagliate relative ad un dato spazio di frequenze. La piramide di tipo *steerable*, tiene conto anche di

informazioni legate all'orientamento specifico delle *feature* di base. Il processo di sintesi partendo da un'immagine contenente *rumore bianco*, via via sintetizza una nuova tessitura, attuando ripetutamente una sorta di equalizzazione degli istogrammi, volta ad avvicinare ai diversi livelli di risoluzione l'istogramma dell'immagine sintetizzata all'istogramma della tessitura di input. A dispetto della relativa semplicità di base dell'algoritmo proposto, l'idea di utilizzare gli istogrammi dei vari livelli della piramide è significativa. Ciascun livello, infatti contiene i dettagli di un ben preciso spazio delle ottave, cioè delle frequenze. Su tale idea di base De Bonet, come descritto nel paragrafo successivo, ha tirato fuori, quello che a tutt'oggi è uno dei più forti risultati nell'ambito della sintesi automatica di tessiture, senza peraltro costruire esplicitamente nessun modello.

L'articolo originale contiene anche una serie di esperimenti esplicativi che mostrano punti di forza e debolezza del metodo proposto. Sono infatti mostrati anche una serie di esperimenti in cui, a causa della natura stessa della tessitura, contenente per esempio dettagli relativamente grandi o orientati in maniera particolare, il processo di sintesi non riesce a riprodurre il processo stocastico di base. Si osservi inoltre come tale approccio da un certo punto di vista fu anche il precursore del filone di ricerca facente capo a Zhu e Mumford. Anche in questo caso infatti si fa un uso estensivo dei cosiddetti istogrammi marginali, che nel modello *FRAME* rappresentano le *feature* statistiche di base.

6. Texture e Multirisoluzione (2): De Bonet

In questo paragrafo si descriverà brevemente l'approccio di De Bonet, che rappresenta attualmente uno dei risultati più forti e comunque sicuramente più

significativi del settore. De Bonet in una serie di lavori (De Bonet - 1997; De Bonet, Viola 1998), ha infatti individuato una tecnica capace di ricostruire stocasticamente una vasta classe di tessiture, senza però fare riferimento ad un esplicito modello probabilistico. Il metodo proposto si basa su un modello per così dire *psicofisico*, che cerca misurare le variazioni percettive utilizzando come discriminanti la risposta a opportuni filtri. Si suppone, inoltre, che se due o più regioni in una data *texture* sono percettualmente indistinguibili, si può riorganizzare la stessa tessitura randomizzandone la loro posizione relativa, preservandone così le caratteristiche percettive legate alla legge stocastica soggiacente la tessitura stessa.

La qualità della tessitura così sintetizzata, può essere stimata, sovrapponendola a quella originale; in tal caso infatti, non se ne dovrebbe scorgere la presenza (effetto *popout*).

Da un punto di vista matematico la procedura può essere rappresentata da una funzione F che prende come argomento la tessitura in input (I_{Input}) e ne restituisce la tessitura sintetizzata (I_{Synth}):

$$F(I_{Input}) = I_{Synth}$$

Le due tessiture non devono essere molto diverse tra di loro:

$$D^*(I_{Input}, I_{Synth}) < T_{\max disc} \quad (1)$$

ove D^* è un'opportuna misura di similarità e $T_{\max disc}$ è una soglia sotto la quale non si discrimina alcuna differenza percettiva tra le tessiture.

Tuttavia esse devono essere differenti per evitare l'effetto "replica":

$$V^*(I_{Input}, I_{Synth}) > T_{\min \text{ diff}} \quad (2)$$

Il successo della tecnica è misurato dall'abilità di minimizzare D^* e massimizzare V^* .

La funzione D^* viene comunque implicitamente definita combinando insieme gli approcci basati sul modello psicofisico e sulla multirisoluzione.

La funzione F viene decomposta in un insieme di funzioni F_i , dove ognuna di esse genera un'ottava dello spazio di frequenza della nuova tessitura I_{Synth} . Il dominio di ciascuna F_i è un sottoinsieme del dominio della F , dove ciascuna F_i ha bisogno delle informazioni contenute negli spazi di frequenza più bassi.

Presa in input la tessitura I_{Input} si eliminano le alte frequenze tramite un filtro passa basso (per esempio Gaussiano) e si ottiene I'_{Input} . Si cerca allora di ottenere da essa una nuova immagine I'_{Synth} . Analogamente si procede ottenendo I''_{Synth} da I''_{Input} senza conoscere I'_{Input} .

Ne segue che la funzione F_i dipende solamente dallo spazio delle frequenze di *input* al di sotto di i :

$$\begin{aligned} F_i(I'_{Input}) &= F_i[L_i(I_{Input}), L_{i+1}(I_{Input}), \dots, L_n(I_{Input})] = \\ &= L_i(I_{Synth}) \end{aligned}$$

dove $L_i(I_{Synth})$ è l' i -esimo livello della piramide Laplaciana.

La funzione originaria F si ottiene combinando le funzioni F_i ai vari livelli, da F_0 fino a F_N . Il problema di sintetizzare una tessitura viene decomposto in una serie di sottoproblemi ai vari livelli di risoluzione.

6.1 Idea di base

L'algoritmo di De Bonet è basato sulla seguente ipotesi strutturale: le tessiture differiscono da altre immagini per il fatto che a certe risoluzioni contengono delle regioni o dettagli, discriminabili percettivamente a meno di una data soglia. Se si restringe opportunamente questa soglia la randomizzazione di queste zone non è tale da alterarne le caratteristiche percettive.

In altre parole, a certe risoluzioni ci sono delle regioni in cui la differenza misurata da D^* , non solo è relativamente bassa ma permette, attraverso la ridistribuzione spaziale di tali regioni, a ciascun livello di risoluzione, di aggiungere dettagli alle frequenze più alte senza da un alto introdurre artefatti percettivi strutturali nella tessitura (D^*) ed incrementandone altresì la differenza visiva (V^*).

La figura che segue mostra ad ogni risoluzione le regioni che è possibile scambiare senza ottenere delle variazioni sul risultato finale.

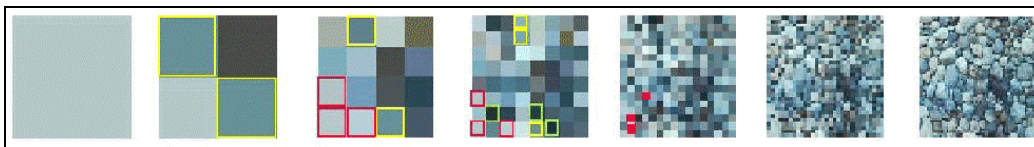


Fig. 4. Regioni *simili* ai diversi livelli di risoluzione.

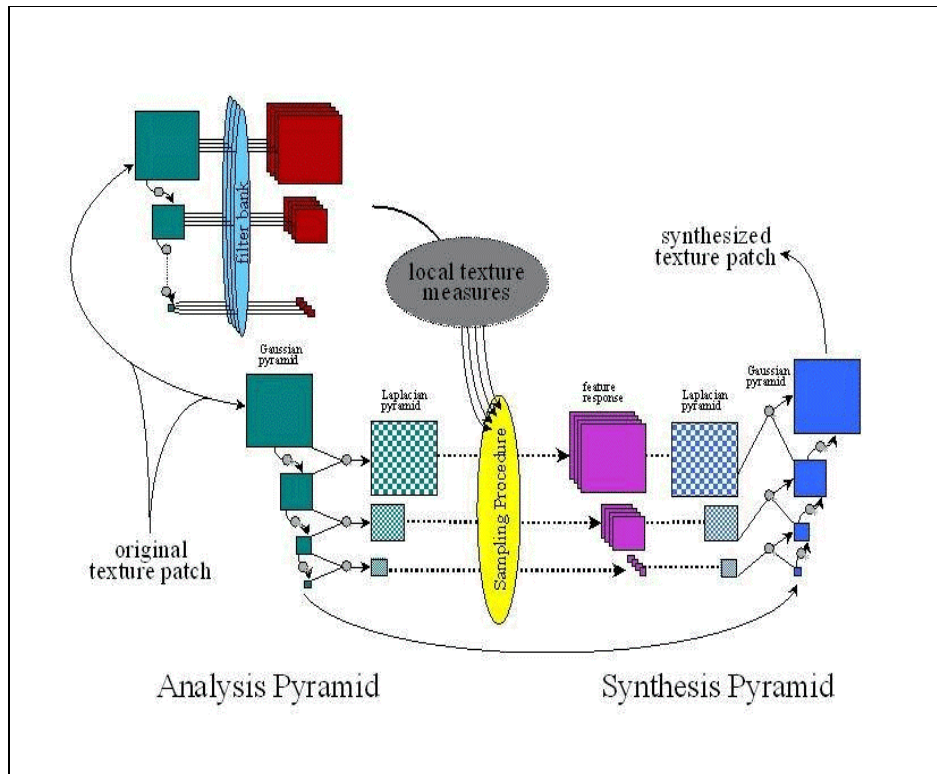


Fig.5. Schema grafico dell'Algoritmo proposto da De Bonet.

6.2 Procedura di De Bonet: Analisi

L'algoritmo in questione procede in due distinte fasi, vedasi Fig.5. Nella prima, detta di analisi, la tessitura viene scomposta in differenti bande di frequenza e ne vengono valutate le caratteristiche tramite l'ausilio di appositi filtri; nella seconda, detta di sintesi, i vari livelli di risoluzione vengono generati sintetizzando campionando opportunamente dal corrispondente livello in input, tenendo conto dei risultati dei filtri. Nella fase di analisi l'immagine viene dunque scomposta nei diversi livelli di risoluzione. Ciò è realizzato utilizzando la decomposizione a piramide Laplaciana (Burt, Adelson - 1993) dove, come già accennato, la frequenza del pixel (x,y) al livello i , nell'immagine I , è data da:

$$L_i(I, x, y) = (G_i(I) - 2 \uparrow [G_{i+1}(I)])(x, y)$$

dove $G_i(I)$ è un livello della piramide Gaussiana ottenuta nel seguente modo:

$$G_i(I) = 2 \downarrow [G_{i-1}(I) \otimes g]$$

dove g rappresenta la maschera di convoluzione Gaussiana e $G_0(I) = I$.

Ogni livello della piramide Laplaciana contiene le informazioni di un determinato spazio di frequenze. Per ogni livello (L_i) di questa piramide Laplaciana ne è sintetizzato uno nuovo (L_i^*) che caratterizzerà la nuova piramide detta di sintesi. Se ciò è fatto indipendentemente ad ogni livello di risoluzione, la immagine sintetizzata non cattura le caratteristiche di quella originale. I valori scelti dovrebbero dipendere non solo dal particolare spazio di frequenze in cui si sta lavorando, ma anche dai valori alle frequenze più basse. Tuttavia l'uso delle informazioni Laplaciane nelle bande di frequenza più basse non è sufficiente, occorre combinarle con le informazioni dei filtri.

6.3 Procedura di De Bonet: Sintesi

Sia $F_i^j(I, x, y)$ la risposta al filtro j -esimo al livello i -esimo del pixel (x, y)

definito come segue:

$$F_i^j(I, x, y) = \begin{cases} (G_i(I) \otimes f_j)(x, y) & \text{Se la dimensione di } G_i(I) \geq f_j \\ 0 & \text{Altrimenti} \end{cases} \quad (3)$$

Si costruisce il vettore $S_i(I, x, y)$ che contiene la storia del *pixel* (x, y) , cioè le risposte ai vari filtri da parte dei *pixel* “genitori” che si trovano nei livelli più bassi.

Pyr-Synt

```

Loop i from top_level-1 down to 0
  Loop (x', y') over Pyr_synth [level i]
    C = 0
    Loop (x, y) over Pyr_analisi [level i]
      C = C ∪ {(x,y)}
      Loop v from top_level down to i+1
        Loop j for each feature
          If D(Pyr_analisi [v][j](x/2v-i, y/2v-i), Pyr_synth [v][j](x'/2v-i, y'/2v-i))
              > threshold[level v][level j]
            then C=C-{(x,y)}
            Break to next (x,y);

Sel = UniformRandom[ 0, ||C||];
(x, y) = C[Sel];
Pyr_synth [v](x', y') = Pyr_analisi [v](x, y);

```

Fig. 6. Algoritmo di Sintesi (De Bonet).

Tali informazioni vengono raccolte in un vettore detto *parent vector* costruito nel seguente modo:

$$S_i(I, x, y) = [F_{i+1}^0 \left(\left\lfloor \frac{x}{2} \right\rfloor, \left\lfloor \frac{y}{2} \right\rfloor \right), \dots, F_{i+1}^M \left(\left\lfloor \frac{x}{2} \right\rfloor, \left\lfloor \frac{y}{2} \right\rfloor \right), \dots, F_N^0 \left(\left\lfloor \frac{x}{2^{N-i}} \right\rfloor, \left\lfloor \frac{y}{2^{N-i}} \right\rfloor \right), \dots, F_N^M \left(\left\lfloor \frac{x}{2^{N-i}} \right\rfloor, \left\lfloor \frac{y}{2^{N-i}} \right\rfloor \right)]$$

dove M rappresenta il numero di filtri utilizzati.

L'algoritmo per la costruzione dei vari livelli L_i^* è sinteticamente descritto in Fig.6. I pixel di L_i^* sono ottenuti nella seguente maniera. Per ogni *pixel* (x', y') in L_i^* si ricava il suo *parent vector* e lo si confronta con quello di tutti i *pixel* (x, y) di L_i . Si costruisce cioè un vettore distanza ottenuto calcolando la distanza

componente per componente. Se tale vettore è più piccolo, elemento per elemento, del vettore soglia T_i , i corrispondenti *pixel* si inseriranno nell'insieme C , dal quale poi campionare i valori RGB da inserire nella posizione in oggetto (x', y') .

In simboli allora:

$$C_i(x', y') = \left\{ (x, y) \mid D(S_i(I_{Synth}, x', y'), S_i(I_{Input}, x, y)) \leq T_i \right\}$$

dove: $T_i = [T_{i+1}^0, T_{i+1}^1, \dots, T_{i+1}^M, \dots, T_N^0, \dots, T_N^M]$. T_i^j rappresenta il livello di soglia per il j -esimo filtro all' i -esimo livello.

L'occhio umano è molto sensibile ai cambiamenti che avvengono a bassa e media frequenza, non lo è altrettanto alle perturbazioni nelle alte frequenze. Facendo riferimento a ciò il valore della soglia è ridotto progressivamente in base al livello in cui ci si trova.

Tale soglia è infatti da intendersi nella seguente forma:

$$T_i^j = \frac{\mathbf{a}}{i^{\mathbf{b}}} \quad (4)$$

dove \mathbf{a} è ottenuto sperimentalmente, mentre $\mathbf{b} \in \{0,1\}$.

E' possibile variare il grado di randomicità delle tessiture sintetizzate manipolando opportunamente il vettore soglia T_i .

4.4 Analisi Computazionale ed Euristiche

L'algoritmo di De Bonet come già detto pone come unico strumento per sintetizzare tessiture reali e/o artificiali la "manipolazione" del vettore delle soglie T .

Ciò significa per esempio, che scegliendo soglie molto basse si vincola il processo di sintesi in maniera forte ottenendo in *output* tessiture molto fedeli all'originale, mentre soglie intermedie consentono alla stessa di avere piccole variazioni.

L'algoritmo di De Bonet almeno nello schema di base suggerito dall'autore, è computazionalmente pesante; ciò è stato confermato anche dalle prove sperimentali realizzate in MatLab 5.0 su un PC Pentium II 350 Mhz, 64MB. Il collo di bottiglia dell'algoritmo è ovviamente rappresentato dal tempo necessario per passare in rassegna, durante la fase di sintesi, tutti i possibili *parent vector* candidati.

Tale complessità computazionale è dovuta al fatto che ad ogni livello, per ogni *pixel* occorre calcolare i *parent vector* di tutti i pixel del corrispondente livello della piramide di analisi. Per diminuire il numero delle computazioni sono state apportate all'algoritmo delle modifiche che trovano fondamento teorico nell'idea che sta alla base dell'algoritmo stesso. La prima di esse riguarda la costruzione dei *parent vector*. L'algoritmo suggerisce di calcolarli a partire dall'ultima componente e di confrontare il valore della distanza con quello della relativa soglia, se è più piccolo si va avanti, altrimenti si interrompe la costruzione e si passa al *pixel* successivo.

Così facendo si risparmiano iterazioni, ma solo al livello in cui mi trovo e non ai livelli successivi. Si noti però che se un dato valore non soddisfa le condizioni di soglia ad un dato livello continuerà ovviamente a non essere adeguato neanche ai livelli superiori.

Una buona euristica sarebbe quella di eliminare dal calcolo tutti quei pixel che condividono quell'elemento nel proprio *parent vector*. Se sono al livello i , al livello $i+1$ tali pixel saranno 4, al livello $i+2$ saranno 8, e in generale al livello $i+h$ saranno 2^{h+1} .

Il tutto è giustificato dal fatto che, in base alla (4), le soglie si assottigliano man mano che si sale di livello e dunque se un valore non va bene quando una soglia è relativamente larga, non può andare bene quando la stessa si restringe.

Tale euristica, che non cambia in alcun modo la qualità della tessitura in *output*, pur apportando notevoli vantaggi in termini di tempo, non risolve del tutto il problema della complessità che, seppur ridimensionato, rimane.

Un'ulteriore modifica apportata all'algoritmo di base è stata implementata osservando la struttura stessa del *parent vector* di un *pixel*.

Tale struttura dati non tiene conto del valore al livello in cui mi trovo, ma solamente dei livelli più alti. Ne segue che ad avere lo stesso *parent vector* ci sono almeno 4 pixel ed essi sono tutti vicini.

In altre parole quando si costruisce il *parent vector* per gli elementi della piramide di sintesi non è necessario farlo per tutti i *pixel*, in quanto a gruppi di quattro si fa riferimento alle stesse coordinate dei livelli precedenti, e, per come è costruito, allo stesso *parent vector*. Tali quattro *pixel* stanno tutti vicini ed è quindi semplice individuarli. Da questa osservazione si deduce che il secondo ciclo dell'algoritmo viene eseguito una volta ogni quattro pixel con conseguente risparmio di tempo.

Anche il terzo *loop* dell'algoritmo può essere eseguito una volta ogni quattro *pixel*, infatti l'osservazione precedente vale anche qui, dato che il *parent vector* è costruito sempre allo stesso modo.

La Figura 7 riporta, a parità di soglia, le sintesi di una data tessitura ottenute facendo uso di differenti versioni dell'algoritmo. La tessitura *a*) è quella di *input*. La *b*) è ottenuta applicando l'algoritmo così come presentato da De Bonet nell'originale articolo del 1997. La *c*) fa uso dell'ottimizzazione che esclude dal calcolo dei *parent vector* dei livelli superiori quelli che non vanno già bene al

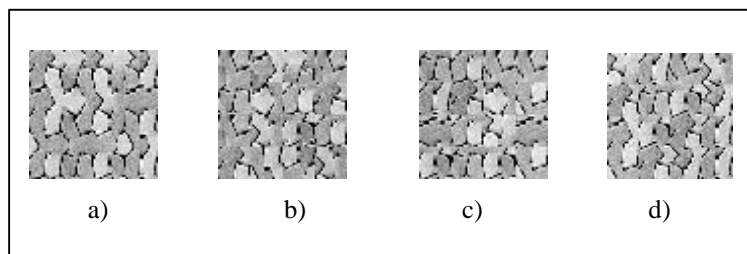


Fig. 7. Applicazione delle diverse euristiche

livello in cui ci si trova. La *d*) sintetizza eseguendo il calcolo dei *parent vector* solo una volta per tutti pixel che lo condividono.

Si può osservare facilmente che i risultati sono soddisfacenti in tutti i casi, ma i tempi impiegati per il calcolo della *b*) 936'' e della *c*) 256'' sono maggiori di quello speso per la *d*) 140''.

Un'altra osservazione riguarda il vettore soglia, ed in particolare la formula (4). Se $\mathbf{b} = 0$ il denominatore è sempre uguale ad uno e dunque la soglia non si restringe mai. Ciò vuol dire che i valori che vanno bene ad un livello andranno bene anche ai livelli più alti. Quando $\mathbf{b} = 1$ la soglia si restringe all'aumentare dei livelli e costringe a scegliere tra valori molto vicini tra di loro, dando vita a

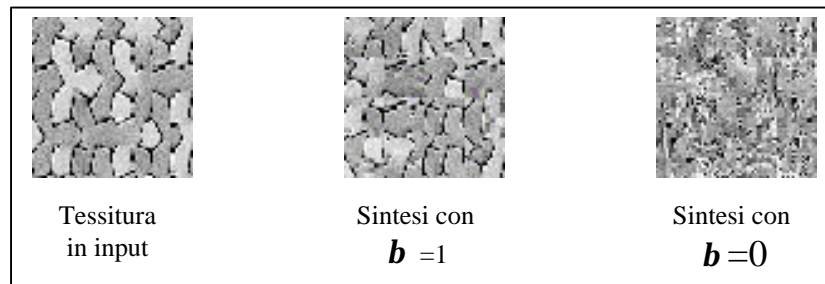


Fig. 8. Differente uso di b nel vettore soglia

tessiture molto più simili all'originale di quanto non lo siano quelle ottenute con la stessa soglia, ma con $b = 0$. Data una tessitura nella Fig. 8 vengono riportati i risultati alla medesima soglia, ma con b differenti. Si può osservare che il cambiamento di tale valore è già determinante per ottenere tessiture più fedeli all'originale.

Poiché i filtri usati sono 5 per indicare il vettore T vengono dati solamente 5 valori (che corrispondono ai vari a), tutti gli altri verranno individuati mediante la formula (4).

E' quindi possibile agire sul vettore soglia T per riuscire a *tarare* il processo di sintesi, anche per tessiture la cui componente deterministica è molto forte.

7. Hypertexture

Un differente approccio legato alla sintesi di superfici cosiddette *textured*, nella rappresentazione di ambienti 3D, è stata introdotta da Perlin e Hoffert nel 1989. Con il termine *Hypertexture*, gli autori indicano una sorta di ibrido fra la tessitura

vera e propria e la cosiddetta *shape*, cioè la forma, l'andamento, le informazioni sulla densità sullo specifico *solido* sul quale la si vuole riprodurre.

La modellizzazione vera e propria si realizza operando con opportune funzioni densità basate appunto sulla forma specifica, distinguendone la regione solida vera e propria, detta *hard*, da quella *soft* dove grazie all'utilizzo combinato di opportuni operatori è possibile generare forme volumetriche specifiche, quali, vetro, pelle, fuoco, ecc.. Più specificatamente è possibile definire una funzione densità-oggetto, *Object Density Function* $D(x)$ e una serie di funzioni f_i dette *Density Modulation Function*, in breve DMF, da utilizzare per controllare le caratteristiche spaziali dell'oggetto. Queste ultime sono quindi legate all'inclinazione, alla presenza di rumore, di turbolenze, ecc.. Le *Hypertexture* sono poi automaticamente generate mediante successive applicazioni di funzioni DMF su un oggetto di densità $D(x)$. La principale innovazione introdotta da Perlin riguarda il fatto che in questo caso si lavora direttamente nello spazio R^3 anzichè su superfici e soprattutto si è così in grado di generare la cosiddetta *apparenza* di fenomeni reali complessi difficilmente ottenibile con i metodi precedentemente esistenti.