

## Vedere Artificială - Tema 3

### Sinteza și transferul texturii

#### Obiectiv:

Scopul acestei teme este implementarea și testarea unei versiuni a algoritmului de sinteză (Figura 1) și transfer (Figura 3) al texturii propus de Alexei Efros și William Freeman în articolul "Image quilting for texture synthesis and transfer" (găsiți articolul atașat în materialele pentru temă).

Funcțiile Matlab care vă vor ajuta la implementarea temei sunt în directorul *cod*; imaginile pe care le veți folosi sunt în directorul *data*.

Pentru a înțelege tema citiți articolul în limba engleză și consultați slide-urile din curs. Toate detaliile legate de predarea temei le găsiți la sfârșitul acestui document.

#### Punct de pornire

Scriptul *ruleazaProiect.m* vă oferă un punct de pornire în implementarea voastră. În acest script se citește imaginea ce urmează a fi sintetizată (eventual și imaginea pe care se transferă

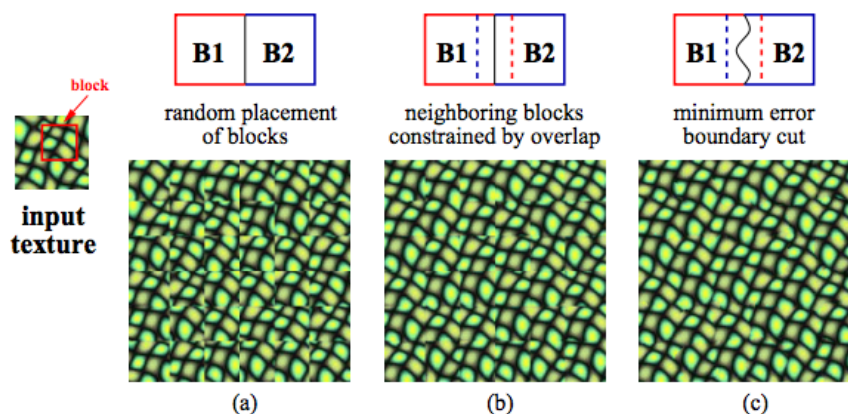


Figura 1: Sinteza texturii. (a) Textura inițială. (b) Textura obținută cu blocuri aleatoare. (c) Textura obținută pe baza erorii de suprapunere. (d) Textura obținută pe baza erorii de suprapunere și a frontierei de cost minim.

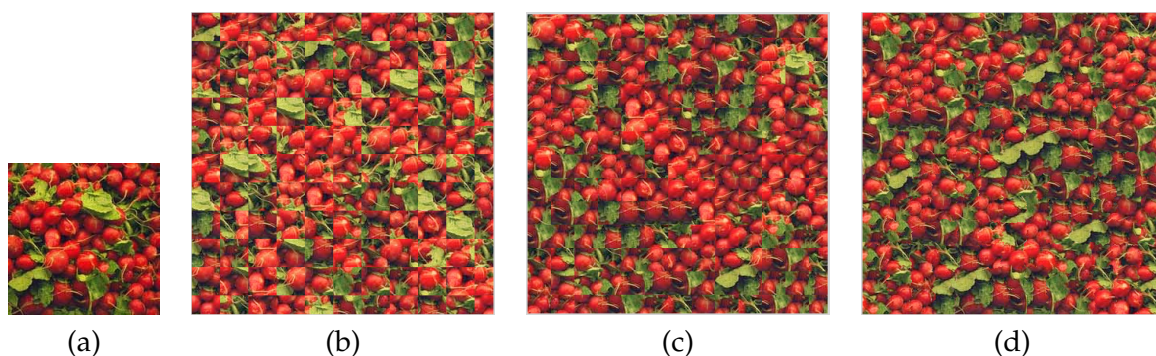


Figura 2: Sinteza texturii. (a) Textura inițială. (b) Textura obținută cu blocuri aleatoare. (c) Textura obținută pe baza erorii de suprapunere. (d) Textura obținută pe baza erorii de suprapunere și a frontierei de cost minim.

textura). Acest script apelează una dintre cele două funcții: *realizeazaSintezaTexturii.m* sau *realizeazaTransferulTexturii.m*.

### 1.1 Sinteza texturii

Implementați mai întâi operația de sintetizare a unei texturi: pornind de la o textură inițială obțineți o imagine de 2 ori mai mare în lățime și lungime care păstrează caracteristicile texturii inițiale. Articolul prezintă 3 modalități de a construi această textură, folosind ideea de a sintetiza blocuri de pixeli. Pornind de la o colecție de blocuri selectate aleator (dimensiunea colecției de blocuri este setată în cod la 2000) la fiecare pas sintetizăm un bloc pe baza blocurilor vecine existente. Cele trei modalități de a sintetiza blocurile prezentate în articol sunt:

- (a) alăturând blocuri selectate în mod aleator. Aceasta opțiune este deja implementată în funcția *realizeazaSintezaTexturii.m*, la fiecare pas un bloc este ales la întâmplare din colecția de blocuri.
- (b) în funcție de eroarea de suprapunere astfel: la fiecare pas se alege din colecție acel bloc care are eroarea de suprapunere mică cu vecinii săi (unul sau doi vecini). Vecinii cu care se suprapune un bloc se definesc astfel: (i) blocurile de pe prima linie au un singur vecin, ele se suprapun la stânga cu blocul vecin; (ii) primul bloc de la linia 2 are un singur vecin, el se suprapune în sus cu primul bloc de pe linia 1; (iii) toate celelalte blocuri au doi vecini, blocul din stânga și de sus. Blocul ales se suprapune întru totul peste vecinul/vecinii săi.
- (c) în funcție de eroarea de suprapunere (ca mai sus) dar de aceasta dată se calculează frontiera de cost minim care desparte blocul ales de vecinii săi. Frontiera de cost minim este drumul (orizontal sau vertical) de cost minim pe porțiunea de suprapunere a doi vecini care uneste o extremitate cu cealaltă. Aceasta frontieră minimă se găsește prin programare dinamică, similar cu cea ați codat la tema anterioară. Blocul ales se suprapune peste vecinul/vecinii în funcție de frontieră găsită (revedeti slide-urile de la curs pentru explicații mai detaliate).

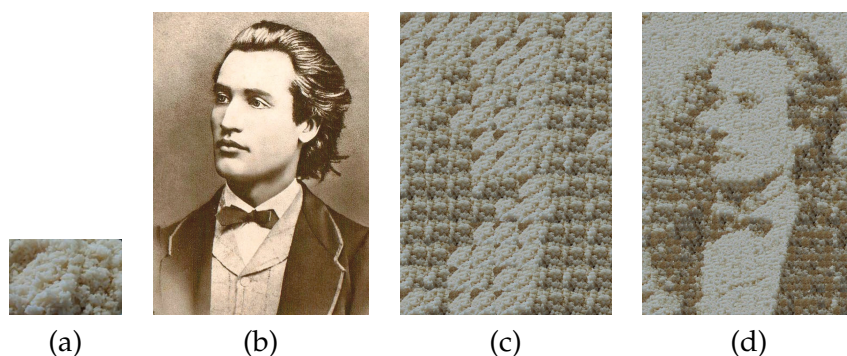


Figura 3: Trasferul texturii. (a) Textura inițială. (b) Imaginea pe care se transferă textura. (c) Rezultatul după iterația 1; (d) Rezultatul după iterația 2.

Exista câțiva parametri care controlează calitatea și rapiditatea realizării sintezei texturii:

- dimensiunea unui bloc pătratic, setată inițial la 36 de pixeli. Practic la fiecare pas se sintetizează  $36 \times 36$  pixeli.
- mărimea colecției de blocuri, setată inițial la 2000 de blocuri.
- porțiunea de suprapunere a două blocuri, setată inițial la  $1/6$  din bloc, adică 6 pixeli.
- eroarea tolerată a distanței celei mai mari față de distanța optimă în momentul în care se caută un bloc care optimizează eroarea de suprapunere cu vecinul/vecinii săi.

## 1.2 Transferul texturii

Secțiunea 3 din articol descrie cum puteți transfera textura pe o imagine (Figura 3). Modificați selecția blocului din colecție astfel încât pe lângă o eroare de suprapunere mică blocul ales trebuie să se potrivească cu intensitatea blocului din imagine pe care se transferă textura. Parametrul  $\alpha$  descris în articol controlează acest trade-off. De asemenea, algoritmul oferă rezultate vizuale mai bune după câteva iterații (detalii în articol).

## 1.3 Predarea proiectului

Puneți într-o arhivă cu numele *tema3\_cod.zip* codul vostru Matlab. Puneți într-un document cu numele *tema3\_rezultate.pdf* următoarele:

- 5 imagini cu texturi sintetizate obținute prin cele 3 opțiuni specificate: aleator, minimizarea erorii de suprapunere, găsirea frontierei de cost minim. Afișați pentru fiecare exemplu textura inițială (puteți folosi texturi din folderul 'data') și rezultatele pentru cele 3 cazuri.
- 2 exemple cu transferuri de textură, afișând textura, imaginea sursă și imaginea rezultată finală (puteți adăga și rezultatele intermediare de la fiecare iterație).

Vom puncta tema în felul următor: 1 punct din oficiu, 6 puncte implementarea corectă a sintetizării texturii (3 puncte pentru fiecare variantă descrisă), 2 puncte transferul texturii, 1 punct prezentarea temei.

Trimiteti cele două fișiere (*tema3\_cod.zip* și *tema3\_rezultate.pdf*) la adresa de email a Iulianei Georgescu, **georgescu\_lily@yahoo.com**.

Termenul limită de predare a proiectului este duminică, 2 decembrie 2018, ora 23:59. Fiecare zi de întârziere în predarea proiectului se penalizează cu 1 punct în minus.