Procedural Computing of String Art

Adrian-Lucian Mincu
Facultatea de Matematică și Informatică

January 31, 2025

Overview

Abstract

Această lucrare de licență abordează tema computației procedurale a ceea ce se numește "String Art", utilizând diverse metode algoritmice. O sursă semnificativă de inspirație a fost lucrarea de cercetare numită "String Art: Towards Computational Fabrication of String Images" [Birsak et al., 2018].

În mod normal, artiștii ar trebui să găsească intuitiv un aranjament al ației care să recreeze o imagine dată. De aceea, o să tratez problema găsirii unei căi matematice de a computa string art, reușind să automatizez procesul.

Obiectivul Lucrării

Scopul intregii lucrări este de a implementa mai multe metode algoritmice care să rezolve această problemă, comparându-le pe toate, astfel putând crea o analiză care să evidențieze metoda cea mai eficientă din punct de vedere al timpului, memoriei, cât și calității vizuale al rezultatului.

String Art

Așa cum am spus mai sus, string art este o tehnică de a crea un efect vizual manipulând ața pentru a forma un model și a reproduce o imagine.
Obiectivul este de a replica acest efect, în mod digital, plecând de la o imagine dată ca input.

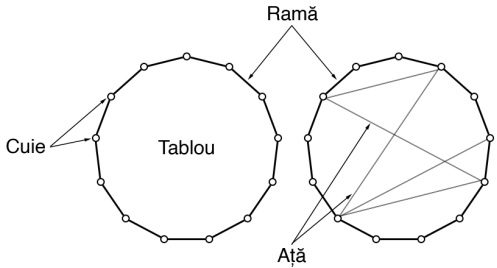

Figure: Imagine care să illustreze fiecare componentă din procesul de string art.

Limitări Fizice

Definiția string art-ului pe care am adresat-o mai devreme nu o să fie aplicată complet când voi recrea imaginile digital, adică unele reguli vor fi relaxate pentru a explora posibilitățile și limitele acestui domeniu.

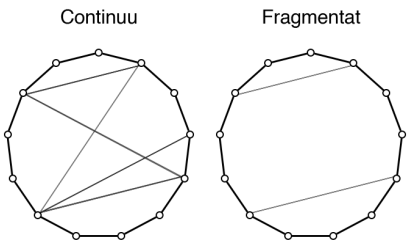

Figure: Imagine care să illustreze diferite tipuri de aranjamente: continuu versus fragmentat.

Terminologie

Putem defini urmatoarele notații:

$$m = \text{Numărul de linii din imagine}$$

$$n = \text{Numărul de coloane din imagine}$$

$$y = \text{Imaginea inițială alb-negru. } y \in \mathbb{R}^{m,\,n}$$

$$Y = \text{Versiunea aplatizată pe rânduri și preprocesată a imaginii. } Y \in [0,1]^{m \cdot n} \subset \mathbb{R}^{m \cdot n}$$

$$p = \text{Numărul de cuie folosite}$$

$$I = \binom{p}{2}\text{, numărul de linii totale care pot fi trase}$$

$$M_i = \text{O matrice care reprezintă a } i\text{-a linie desenată pe tablou}$$

Terminologie

$M =$ Pentru fiecare M_i, coloana corespunzătoare este versiunea aplatizată pe rânduri a M_i.
$M \in \mathbb{R}^{m \cdot n,\ l}$

Vectorul coloană X care este output-ul algoritmului nostru, unde fiecare element din el, x_i reprezintă dacă linia i va fi trasată sau nu.

$$x_i = \begin{cases} 1 & \text{dacă linia } i \text{ este trasată} \\ 0 & \text{altfel} \end{cases}, \quad X \in \mathbb{R}^l$$

Funcția Obiectiv

Astfel, obiectivul nostru final este să aflăm valorile elementelor vectorului X care ne aduc cel mai aproape de imaginea inițială. Deci trebuie să minimizăm:

$$\min \| \sum (M_i \cdot x_i) - y \|$$

sau

$$\min \| M \cdot X - Y \|$$

Calitatea Subiectivă a Imaginii

Este important de menționat că există o discrepanță între modul în care calculatorul percepe o imagine calitativă și modul în care acesta este percepută de ochiul uman. În formula de mai sus, calculatorul va considera că imaginea cu cea mai mică eroare este cea de cea mai înaltă calitate.

i03_24_5, MOS=0.4, FSIMc=0.693 i03_13_4, MOS=2.9, FSIMc=0.671

Fig. 27. Example of contradiction between FSIMc and MOS for the test image # 25.

Figure: Exemplu din lucrarea [et al., 2015] care evidențiază discrepanța între erorile computate de calculator și calitatea percepută.

Calitatea Subiectivă a Imaginii

Acest fenomen poate fi observat în imaginea de mai sus 3, unde poza din stânga, deși are un scor FSIMc [Zhang et al.,] mai mare (un scor mai mare indicând teoretic o calitate mai bună) scorul MOS [1] este mai mic.

Acest lucru înseamnă că majoritatea oamenilor au perceput imaginea din dreapta ca fiind mai clară și mai calitativă, contrar evaluării calculate algoritmic. Trebuie menționat că FSIMc este o metodă mai avansată decât MSE (folosită în această lucrare) și, cu toate acestea, poate să dea greș în evaluarea calității.

[1] Pentru detalii suplimentare despre Scorul Mediu de Opinie (MOS), consultați articolul dedicat Mean Opinion Score (MOS)

Plasarea Cuielor

Pentru a plasa cuiele echidistant unul față de celălalt pe o formă circulară, am utilizat ecuația parametrică a cercului.

Pentru un cerc centrat in (x_c, y_c) cu rază r și p cuie:

$$x_k = x_c + r \cdot \cos \frac{2\pi k}{p}$$

$$y_k = y_c + r \cdot \sin \frac{2\pi k}{p}$$

$$\text{for } k = 0, 1, 2, ..., p - 1$$

Trasarea Liniilor

Pentru a trasa linii în reprezentarea matricială, algoritmul Bresenham [2] a fost folosit, determinând punctele necesare care să fie selectate între două cuie alese pentru a avea o reprezentare apropriată de o linie dreaptă desenată.

[2] Pentru implementarea detaliată a algoritmului, consultați articolul despre Bresenham's Line Algorithm

Obținerea Rezultatului

După obținerea vectorului X, imaginea finală este generată înmulțind matricea de linii cu vectorul coloană X, rezultând combinațiile de linii care trebuie trasate: $O = MX$. Totuși, valorile din O pot depăși intervalul [0, 1].

Pentru a rezolva acestă problemă trebuie introdusă o funcție de limitare [Socha, 2024]. $C = \mathbb{R}^+ \to [0, 1]$. $C(x) = \max(0, \min(x, 1))$. În plus, pentru a desena imaginea cu culoarea neagră, trebuie inversate valorile culorilor. În final, acestea sunt scalate în intervalul [0, 255], specific imaginilor digitale. Formula completă devine:

$$O = (1 - C(MX)) \cdot 255$$

Metoda celor Mai Mici Pătrate (CMMP)

O primă metodă, care a fost inspirată din videoclipul [3] este cea folosind **CMMP**.
Metoda CMMP minimizează funcția menționată mai sus: $\min \|M \cdot X - Y\|$ prin vectorul X astfel încât norma l_2 dintre imaginea originală și cea rezultată să fie minimă.

Metoda CMMP poate fi implementată prin două abordări.

- Reprezentare densă a matricei.
- Reprezentare rară a matricei.

[3] Inspirația pentru această metodă poate fi găsită în videoclipul The Mathematics of String Art

Metoda celor Mai Mici Pătrate (CMMP) Densă

În acestă abordare, matricea M este reprezentată ca o matrice densă (plină), cu dimensiunile: $M \in \mathbb{R}^{m \cdot n,\ 1}$.
Deși această abordare este simplă și ușor de înțeles, reprezentarea densă este mult mai costisitoare din punct de vedere computațional pentru matrici mai mari, deoarece fiecare element, incluzând și zero-urile, este stocat explicit.

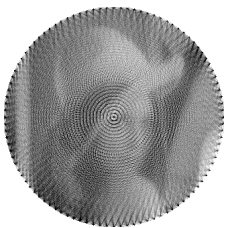

Figure: Imagine rezultată în urma procesului de CMMP cu reprezentare densă a matricii.

Metoda celor Mai Mici Pătrate (CMMP) Rară

A doua abordare este cea cu reprezentare rară a matricilor [4].

Știind faptul că vectorii noștri coloană din reprezentarea densă a matricei constituie o matrice aplatizată pe rânduri în care o singură linie a fost trasă, atunci putem spune că matricea este una rară, însemnând că majoritatae elementelor sunt zero.

În acest caz, algoritmii CMMP au implementări speciale [csr,] care să profite de structura rară a matricii. Sunt mai rapizi și folosesc mai puțină memorie pentru că procesează doar elementele diferite de zero.

[4] Pentru detalii despre reprezentarea rară a matricilor, consultați Sparse Matrix

Metoda celor Mai Mici Pătrate (CMMP) Rară

De exemplu, cel mult $\max(m, n)$ elemente ale matricei M_i au valori diferite de 0, datorită algoritmului Bresenham. Observați imaginea de mai jos.

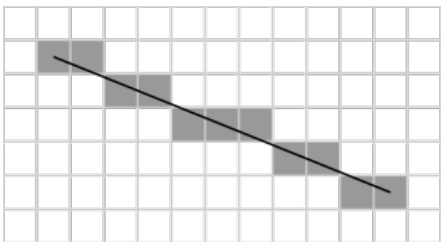

Figure: Imagine care să demonstreze modul de desenare al algoritmului Bresenham.

În exemplul de mai sus, unde $m = 5,\ n = 1$, linia trasată are exact 11 elemente diferite de zero.

Metoda celor Mai Mici Pătrate (CMMP) Rară

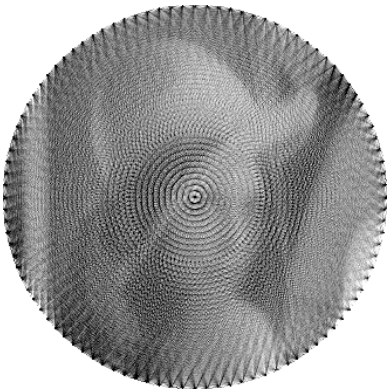

Figure: Imagine rezultată în urma procesului de CMMP cu reprezentare rară a matricii.

Matching Pursuit Greedy

Metoda Greedy constă în alegerea unei linii la fiecare pas. Această alegere este făcută astfel încât să minimizeze eroarea dintre imaginea originală si imaginea curentă la pasul k.

Având în vedere timpul de computare mare, cauzat de necesitatea de a rezolva o problemă de tip CMMP la fiecare pas iterativ, am decis să integrez două euristici:

1. Euristică Random

   Procesul este simplu, se selectează random k linii candidat.

2. Euristică Produs Scalar

   În acest caz se computează MY și se aleg primele k linii care nu au fost desenate încă și au cel mai mare produs scalar.

Metoda Greedy Rezultate

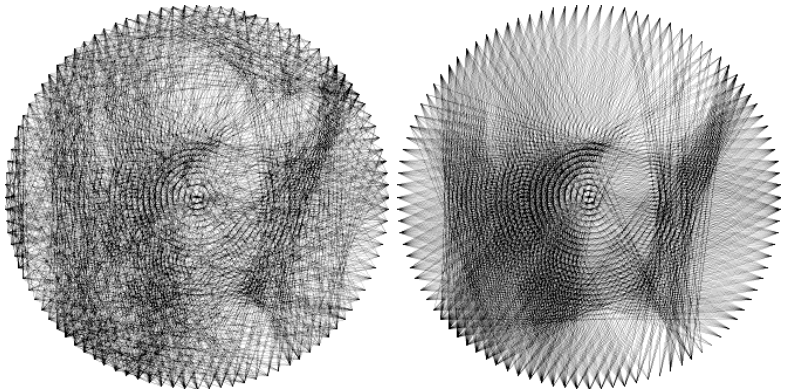

Figure: Imagini rezultate în urma procesului de Greedy cu euristică random (stânga) și euristică produs scalar (dreapta) cu 1000 de linii selectate.

Orthogonal Matching Pursuit (OMP)

Metoda Orthogonal Matching Pursuit (OMP) este foarte similară cu cea Greedy, având în procesul iterativ același algoritm bazat pe selecția liniei care reduce cel mai mult eroarea.

Spre deosebire de metoda Greedy, OMP actualizează reziduul Y (inițial fiind imaginea originală) prin scăderea proiecției liniei selectate.

Astfel se garantează ca reziduul rămas este ortgonal față de liniile alese anterior.

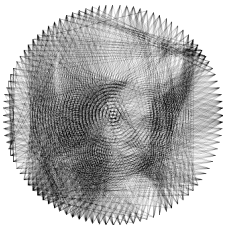

Figure: Imagine rezultată în urma procesului de Orthogonal Matching Pursuit (OMP) cu 1000 de linii selectate.

Input

Figure: Imaginea originală.

Rezultate CMMP

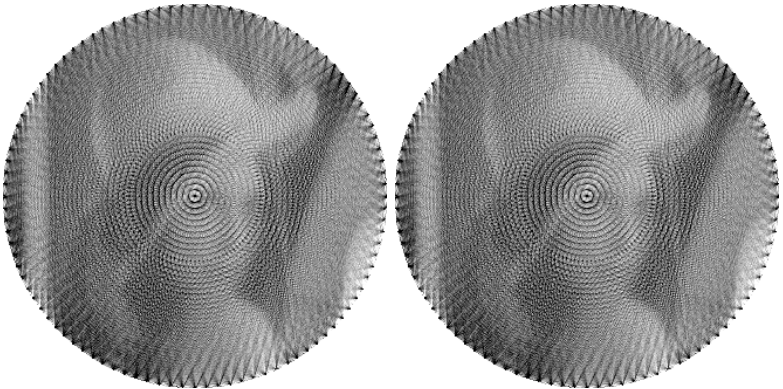

Figure: Imagini rezultate în urma procesului de CMMP cu reprezentări matriciale dense (stânga) și rare (dreapta).

Rezultate Matching Pursuit

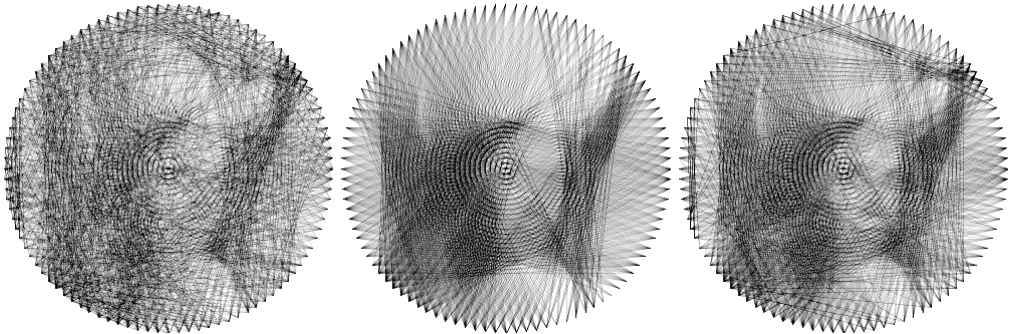

Figure: Imagini rezultate în urma procesului de Greedy cu euristică random (stânga) și euristică produs scalar (mijloc) și OMP (dreapta), cu 1000 de linii selectate.

Observații vizuale

Se poate observa încă de la început că cele mai bune rezultate pentru ochiul uman sunt cele care folosesc metoda CMMP. Acest lucru se întâmplă din cauză că ele utilizeaza toate liniile posibile și ajustează intensitatea lor între 0 (alb) și 1 (negru).

Pe de altă parte, abordările Greedy, care selectează un număr limitat de linii (în acest caz, 1000), produc reconstrucții mai puțin netede. Euristica random (stânga) tinde să fie mai haotică, alegând linii din diferite regiuni ale imaginii. În schimb, euristica produsului scalar (mijloc) se concentrează pe anumite zone, ducând la rezultate mai localizate, dar mai puțin variate.

Metoda OMP pare să fie foarte apropriată cu cea de Greedy care folosește euristica produs scalar. Doar că din cauza diferențelor în implementare, se poate observa că OMP are o alegere mai diversă, mai punctual, selectează și linii de pe margine.

Imagini Diferență

Imaginile diferență evidențiază discrepanțele dintre imaginea originală și rezultatul fiecărei metode.

Figure: Diferența imaginilor rezultate până la imaginea originală.

Timp de Computare

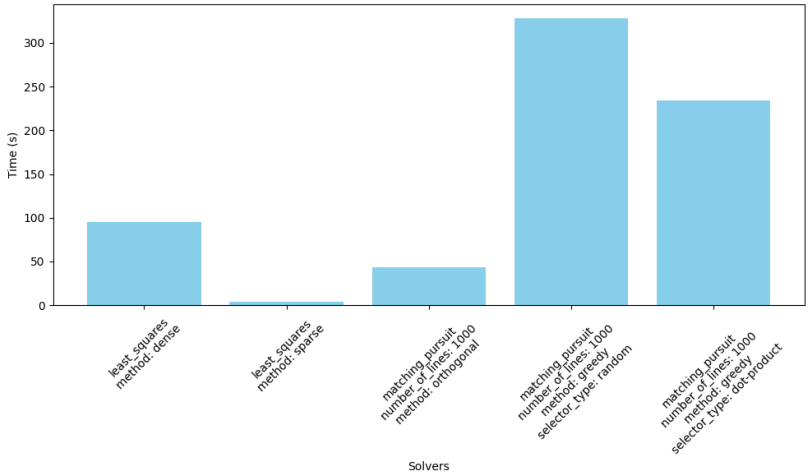

Figure: Imagine care ilustrează timpul necesar pentru computarea rezultatului în funcție de fiecare metodă.

Timp de Computare

În ceea ce privește timpul de computare, metoda CMMP rară este cea mai eficientă, finalizându-se în doar câteva secunde. Metoda CMMP densă, deși necesită mai mult timp, rămâne relativ rapidă în comparație cu algoritmii matching pursuit.

Metodele Greedy, în special euristica random, prezintă cele mai lungi timpuri de computare. Acest lucru se datorează faptului că acestea trebuie să calculeze repetat soluția de tip CMMP pentru fiecare linie candidat și să o selecteze pe cea care minimizează eroarea la fiecare pas.

Datorită metodei mai eficiente de selectare a liniei din cadrul abordării OMP, se observă că acest algoritm este semnificativ mai rapid decât metoda Greedy și, surprinzător, mai rapid decât CMMP dens. Cu toate acestea, OMP nu este încă la fel de rapid ca CMMP rar.

Utilizarea Maximă a Memoriei

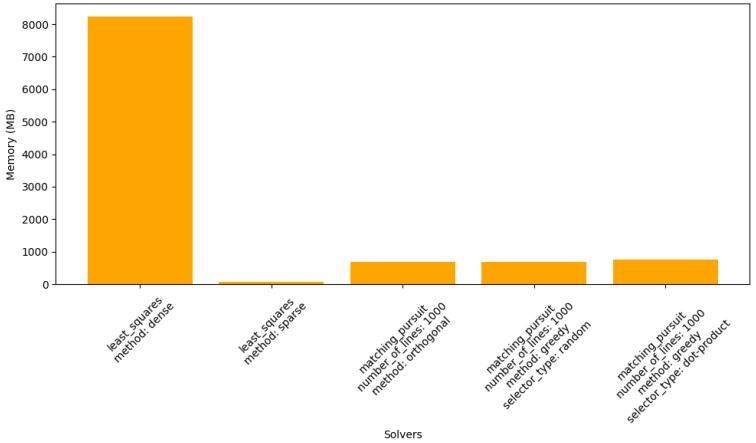

Figure: Imagine care ilustrează memoria maximă utilizată în timpul computării rezultatului în funcție de fiecare metodă.

Utilizarea Maximă a Memoriei

Utilizarea maximă a memoriei oferă o perspectivă interesantă asupra eficienței algoritmilor. Toate metodele, cu excepția abordării CMMP densă, utilizează mai puțin de 1GB de memorie.

Metoda densă necesită mult mai multă memorie deoarece trebuie să stocheze o matrice mare de dimensiune: $m = 108.900,\ n = 4.950$. În schimb, metoda rară trebuie doar să stocheze elementele nenule, reprezentând o soluție mai eficientă din punct de vedere al memoriei.

Iar abordările OMP și Greedy păstrează, în cel mai rău caz, o matrice de dimensiune: $m = 108.900, n = 1000$. Unde 1000 e numarul de linii care trebuie desenate.

Concluzii

Această analiză evidențiază trade-off-urile între precizie, timp de calcul și eficiență a memoriei.

Cea mai bună metodă de până acum este abordarea CMMP cu reprezentare rară a matricei. Aceasta se remarcă prin mai multe motive:

- Timp de computare cel mai scăzut
- Utilizare memorie minimă
- Cel mai mic RMS
- Cea mai bună calitate subiectivă a imaginii

În general, abordarea CMMP cu reprezentare rară a matricei atinge un echilibru optim între performanță și eficiență, făcând-o soluția cea mai fiabilă în această evaluare de performanță.

Bibliografie

csr_matrix.
Accessed: 2025-01-16.

Birsak, M., Rist, F., Wonka, P., and Musialski, P. (2018).
String art: Towards computational fabrication of string images.
ResearchGate.

et al., N. P. (2015).
Image database tid2013: Peculiarities, results and perspectives.
ScienceDirect, page 76.

Socha, R. (2024).
Computational string art.
KTH, page 5.

Zhang, L., Mou, X., and Zhang, D.
Fsim: A feature similarity index for image quality assessment.
PolyU.