LAPORAN TEKNIS

TUGAS KELOMPOK ANALISIS NUMERIK

Studi Penggunaan Bézier Curves dalam Pembuatan Ilustrasi Gambar yang Sudah Ada

Disusun oleh Kelompok C10:

Argya Farel Kasyara	2306152424
Alexander William Lim	2306207505
Jason Kent Winata	2206081313
Daffa Abhipraya Putra	2306245131

Fakultas Ilmu Komputer
Universitas Indonesia

2025

"Dengan ini, kami menyatakan bahwa tugas ini adalah hasil pekerjaan kelompok sendiri."

Argya Farel Kasyara

NPM: 2306152424

Alexander William Lim

NPM: 2306207505

Jason Kent Winata

NPM: 2206081313

Daffa Abhipraya Putra

NPM: 2306245131

Rangkuman

Laporan ini membahas implementasi metode numerik untuk merepresentasikan gambar raster menjadi ilustrasi vektor menggunakan kurva Bézier kubik. Fokus utama laporan ini adalah metode **Pure Interpolation**, di mana setiap titik kontur yang diekstraksi dari citra raster dihubungkan secara langsung menggunakan kurva Bézier. Selain itu, kami juga membandingkan metode ini dengan pendekatan optimisasi menggunakan *Least Squares Fitting* untuk menganalisis perbedaan efisiensi dan kualitas representasi.

Daftar Isi

1. Pendahuluan

1.1 Latar Belakang

Kurva Bézier merupakan salah satu primitif grafis terpenting dalam *Computer Assisted Design* (CAD) dan grafis vektor. Kemampuannya untuk memodelkan bentuk lengkung yang halus dengan hanya beberapa titik kontrol menjadikannya sangat efisien dibandingkan dengan representasi poligon atau raster. Dalam proyek ini, kami mengeksplorasi bagaimana mengonversi citra raster statis menjadi representasi matematis menggunakan kurva Bézier, sebuah proses yang dikenal sebagai *vectorization* atau *image tracing*.

1.2 Rumusan Masalah

Permasalahan utama yang dikaji adalah:

1. Bagaimana mengekstraksi titik-titik data yang merepresentasikan bentuk dari sebuah citra raster?
2. Bagaimana merepresentasikan sekumpulan titik data tersebut menggunakan kurva Bézier dengan metode interpolasi murni?
3. Bagaimana membandingkan efisiensi metode interpolasi murni dengan metode optimisasi *Least Squares*?
4. Bagaimana cara menghasilkan berkas keluaran standar (PDF) yang dapat memvisualisasikan hasil kurva tersebut?

1.3 Tujuan Penulisan

1. Mengekstraksi titik-titik data kontur dari citra raster.
2. Mengimplementasikan algoritma *Pure Interpolation* untuk menghubungkan setiap titik kontur dengan kurva Bézier.
3. Menganalisis perbandingan antara metode *Pure Interpolation* dan *Least Squares Fitting*.
4. Menghasilkan berkas PDF vektor dari hasil perhitungan kurva.

1.4 Batasan Masalah

- Kurva yang digunakan adalah Bézier kubik (derajat 3).
- Masukan berupa citra raster (PNG/JPG).
- Keluaran berupa berkas PDF.

2. Dasar Teori

2.1 Dasar Teori

Kurva Bézier derajat n didefinisikan oleh persamaan parametrik menggunakan polinomial Bernstein:

$$B(t) = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} t^i P_i, \quad 0 \leq t \leq 1$$

Untuk kasus Cubic Bézier ($n = 3$), persamaannya adalah:

$$B(t) = (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t) t^2 P_2 + t^3 P_3$$

Representasi ini menjamin bahwa kurva dimulai di P_0 (saat $t = 0$) dan berakhir di P_3 (saat $t = 1$). Garis singgung di P_0 searah dengan vektor $\vec{P_0 P_1}$, dan di P_3 searah dengan $\vec{P_2 P_3}$.

2.2 Pure Interpolation

Dalam metode *Pure Interpolation*, tujuan utamanya adalah membuat kurva yang melewati setiap titik data yang tersedia. Jika diberikan sekumpulan titik $D_0, D_1, \ldots, D_k$, kita membuat segmen kurva Bézier antara setiap pasangan titik berurutan D_i dan D_{i+1}.

Untuk memastikan kontinuitas C^1 (kemulusan sambungan), titik kontrol P_1 dan P_2 harus dipilih sedemikian rupa sehingga garis singgung masuk dan keluar selaras. Namun, untuk implementasi dasar yang menghubungkan setiap titik (piecewise linear approximation using Bezier), kita dapat menempatkan titik kontrol pada garis lurus antara D_i dan D_{i+1}:

$$P_1 = D_i + \frac{1}{3}(D_{i+1} - D_i)$$
$$P_2 = D_i + \frac{2}{3}(D_{i+1} - D_i)$$

Ini secara efektif membuat kurva Bézier berperilaku sebagai garis lurus yang menghubungkan D_i dan D_{i+1}.

3. Metodologi Eksperimen

3.1 Pengolahan Citra (Image Processing)

Kami menggunakan pustaka `cv2` (OpenCV) untuk tahap pra-pemrosesan. Kami menggunakan *Canny Edge Detection* untuk mendeteksi tepi berdasarkan gradien intensitas, sehingga mampu menangkap kontur logo dengan akurat. Hasil dari tahap ini adalah sekumpulan titik koordinat (x, y) yang merepresentasikan garis tepi gambar.

3.2 Pure Interpolation Algorithm

Algoritma utama kami untuk metode ini (`fit_curve_pure_interpolation`) sangat sederhana namun menjamin akurasi posisi absolut karena kurva melewati setiap titik yang terdeteksi.

Langkah-langkah: 1. Terima masukan berupa daftar titik kontur $D_0, D_1, ..., D_k$. 2. Iterasi dari $i = 0$ sampai $k - 1$. 3. Tentukan $P_0 = D_i$ dan $P_3 = D_{i+1}$. 4. Hitung titik kontrol internal P_1 dan P_2 menggunakan interpolasi linear (seperti dijelaskan di Dasar Teori). 5. Simpan segmen kurva $[P_0, P_1, P_2, P_3]$.

Metode ini menghasilkan jumlah kurva yang sama dengan jumlah segmen garis antar titik (jumlah titik - 1).

3.3 Pembuatan PDF

Kami mengembangkan generator PDF (`pdf_generator.py`) yang menulis perintah-perintah grafis PDF secara manual.

Pseudocode:

```
Algorithm: Generate PDF
Input: List of Bezier curves, Output filename

1. Initialize PDF Structure:
   Write the PDF Header ("%PDF-1.4").

2. Construct Content Stream:
   For each curve in the list:
       Append "Move To" command (m) for P0.
       Append "Curve To" command (c) using P1, P2, P3.
       Append "Stroke" command (S).

3. Finalize PDF:
   Write Object Catalog, Pages, and Page objects.
   Write Cross-Reference Table (xref).
   Write Trailer and EOF.
```

4. Hasil dan Analisis (Pure Interpolation)

Pada bagian ini, kami menyajikan hasil eksperimen menggunakan metode **Pure Interpolation**.

I. Data Eksperimen

Kami melakukan percobaan pada gambar logo Makara UI (`makara.png`). Berikut adalah statistik hasil pemrosesan:

Tabel 4.1 Statistik Pure Interpolation pada Makara UI

Metrik	Nilai
Dimensi Gambar	1890×2065 piksel
Total Kontur	46
Total Titik Asli	74,615 titik
Total Kurva Bézier	**74,569 kurva**
Rasio Kompresi	**1 kurva per 1.00 titik**

II. Visualisasi Hasil

Input	Output (Pure Interpolation)
`../input/makara.png`	[IMAGE: Hasil PDF Pure Interpolation]

Gambar 4.1 Perbandingan Masukan dan Keluaran Pure Interpolation

III. Analisis

1. **Akurasi Sempurna:** Karena setiap titik kontur dihubungkan secara langsung, hasil visualisasi sangat akurat dan tidak ada detail yang hilang. Kurva mengikuti setiap piksel dari tepi yang terdeteksi oleh Canny Edge Detection.
2. **Inefisiensi Ukuran:** Kelemahan utama metode ini terlihat jelas pada jumlah kurva yang dihasilkan. Dengan 74,615 titik asli, dihasilkan 74,569 kurva Bézier. Ini berarti tidak ada kompresi data sama sekali (rasio 1:1).
3. **Ukuran File:** File PDF yang dihasilkan akan memiliki ukuran yang sangat besar karena harus menyimpan definisi untuk puluhan ribu kurva, yang sebenarnya bisa direpresentasikan dengan lebih sedikit kurva jika menggunakan aproksimasi.

5. Menggunakan Least Square untuk Optimisasi

Sebagai perbandingan dan solusi atas inefisiensi metode *Pure Interpolation*, kami juga mengimplementasikan metode **Hybrid** yang menggunakan **Least Squares Fitting**.

5.1 Metodologi (Recursive Curve Fitting)

Algoritma ini bekerja secara rekursif: 1. Mencoba mencocokkan *satu* kurva Bézier pada sekumpulan titik kontur menggunakan metode kuadrat terkecil (*Least Squares*). 2. Menghitung galat maksimum antara kurva hasil fitting dengan titik asli. 3. Jika galat melebihi ambang batas (*threshold*), segmen titik dibagi dua pada titik dengan galat terbesar, dan proses diulang untuk kedua bagian tersebut.

5.2 Hasil Eksperimen (Least Squares)

Kami menjalankan algoritma ini pada gambar yang sama (`makara.png`) untuk membandingkan hasilnya.

Tabel 5.1 Statistik Least Squares pada Makara UI

Metrik	Nilai
Total Titik Asli	74,615 titik
Total Kurva Bézier	**797 kurva**
Rasio Kompresi	**1 kurva per 93.62 titik**

5.3 Perbandingan Visual dan Efisiensi

Output (Pure Interpolation)	Output (Least Squares / Hybrid)
[IMAGE: Hasil PDF Pure Interpolation]	[IMAGE: Hasil PDF Least Squares]
74,569 Kurva	**797 Kurva**

Gambar 5.1 Perbandingan Visual Pure Interpolation vs Least Squares

Analisis Perbandingan: * **Efisiensi Ekstrem:** Metode *Least Squares* mampu mereduksi jumlah kurva dari **74,569** menjadi hanya **797**. Ini adalah pengurangan sebesar **98.9%** dalam jumlah primitif grafis yang diperlukan. * **Kualitas Visual:** Secara visual, hasil *Least Squares* tetap terlihat sangat mulus dan mempertahankan bentuk asli logo dengan baik. Mata manusia sulit membedakan perbedaan antara interpolasi murni setiap piksel dengan aproksimasi kurva halus, terutama pada resolusi tinggi. * **Kesimpulan:** Metode *Least Squares* jauh lebih superior untuk keperluan ilustrasi vektor karena menghasilkan file yang ringan dan efisien tanpa mengorbankan kualitas visual secara signifikan.

6. Kesimpulan

Proyek ini berhasil mendemonstrasikan dua pendekatan dalam vektorisasi citra menggunakan kurva Bézier:

1. **Pure Interpolation:** Menghubungkan setiap titik data secara langsung. Metode ini menjamin akurasi 100% terhadap data input tetapi sangat tidak efisien (rasio kompresi 1:1), menghasilkan puluhan ribu kurva untuk gambar kompleks.
2. **Least Squares Fitting (Optimisasi):** Menggunakan pendekatan statistik untuk mencocokkan kurva terbaik pada sekumpulan titik. Metode ini terbukti sangat efektif, mampu merepresentasikan gambar yang sama dengan jumlah kurva yang jauh lebih sedikit (rasio kompresi ~1:93) sambil mempertahankan kualitas visual yang tinggi.

Untuk aplikasi praktis seperti desain grafis dan penyimpanan aset digital, metode optimisasi dengan *Least Squares* adalah pilihan yang jauh lebih baik dibandingkan interpolasi murni.

Daftar Referensi

Sauer, T. (2012). *Numerical Analysis* (2nd ed.). Pearson.

Lampiran

Complete Source Code: https://github.com/absolutepraya/bezier-curves-python