

Implementasi Algoritma Scale-Invariant Feature Transform (SIFT) untuk Pembuatan Panorama

Rizky Cahyono Putra
Teknik Informatika
Universitas Darussalam Gontor
Ponorogo, Indonesia
NIM: 442023611012

Abstract—Laporan ini memaparkan implementasi algoritma Scale-Invariant Feature Transform (SIFT) untuk menyelesaikan masalah pencocokan fitur dan penyusunan panorama citra. Eksperimen dilakukan dalam empat tahapan utama: deteksi keypoint, visualisasi konsep Difference of Gaussian (DoG), pencocokan fitur (feature matching) menggunakan metode Brute-Force dengan Lowe's Ratio Test, dan penggabungan citra (image stitching). Hasil percobaan menunjukkan bahwa SIFT mampu mendeteksi fitur yang invarian terhadap skala dan rotasi, serta berhasil menyusun panorama dari dua citra yang memiliki sudut pandang berbeda namun saling tumpang tindih (overlap). Kode sumber lengkap tersedia di: github.com/rizkycahyono97/visikom.

Index Terms—Computer Vision, SIFT, Feature Matching, Panorama, Homography.

I. PENDAHULUAN

Visi komputer bertujuan untuk meniru kemampuan persepsi manusia dalam memahami informasi visual. Salah satu tantangan utama dalam visi komputer adalah mengenali objek atau fitur yang sama pada citra yang memiliki perbedaan skala, rotasi, atau pencahayaan. Algoritma *Scale-Invariant Feature Transform* (SIFT) yang diperkenalkan oleh David Lowe [1] merupakan metode yang populer untuk mendeteksi dan mendeskripsikan fitur lokal pada citra.

Tujuan dari tugas besar ini adalah:

- Memahami konsep dasar *Keypoint* dan *Descriptor*.
- Menganalisis peran *Scale Space* dan *Difference of Gaussian* (DoG).
- Melakukan pencocokan fitur antar citra.
- Mengimplementasikan teknik *Homography* untuk membuat panorama.

II. METODOLOGI DAN ANALISIS

A. Bagian A: Visualisasi Keypoint SIFT

Pada tahap ini, dilakukan deteksi keypoint pada citra *grayscale*. Fungsi `detectAndCompute` dari pustaka OpenCV digunakan untuk mendapatkan lokasi keypoint dan vektor deskriptornya.

Hasil visualisasi (Gambar 1) menampilkan indikator visual berupa lingkaran dan garis:

- **Lingkaran:** Merepresentasikan *scale* (skala) dari fitur tersebut. Semakin besar lingkaran, semakin besar area fitur yang dideteksi (misalnya, gumpalan awan vs kerikil kecil).

- **Garis (Panah):** Merepresentasikan *orientation* (orientasi) dominan dari gradien piksel di sekitar titik tersebut. Hal ini membuat SIFT tahan (*invariant*) terhadap rotasi gambar.



Fig. 1. Visualisasi Keypoint SIFT dengan indikator skala dan orientasi.

B. Bagian B: Scale Space dan Difference of Gaussian

SIFT menggunakan ruang skala (*Scale Space*) untuk mendeteksi fitur yang relevan pada berbagai ukuran. Untuk menyimulasikan ini, dilakukan operasi Gaussian Blur dengan σ_1 dan σ_2 , kemudian dihitung selisihnya menggunakan *Difference of Gaussian* (DoG).

Analisis: Mengapa DoG? SIFT menggunakan DoG sebagai pendekatan (aproksimasi) dari *Laplacian of Gaussian* (LoG). Menghitung LoG secara langsung sangat berat secara komputasi. DoG memberikan hasil yang hampir sama efektifnya dalam mendeteksi tepi dan sudut (blob detection) tetapi dengan biaya komputasi yang jauh lebih efisien. Gambar 2 menunjukkan bahwa DoG berhasil menonjolkan tepi objek dan mengabaikan area yang datar (homogen).



Fig. 2. Visualisasi Gaussian Blur dan Difference of Gaussian (DoG).

C. Bagian C: Feature Matching

Pencocokan fitur dilakukan menggunakan `BFMatcher` (Brute-Force) dengan metode K-Nearest Neighbor ($k = 2$). Untuk menyaring pasangan yang salah (*false positives*), diterapkan *Lowe's Ratio Test*.

Analisis Ratio Threshold: Dalam percobaan ini digunakan threshold sebesar 0.75.

- Jika rasio jarak antara tetangga terdekat pertama dan kedua < 0.75 , maka fitur dianggap unik dan valid (*Good Match*).
- Pengaruh nilai threshold: Jika nilai threshold dinaikkan (misal ke 0.9), jumlah match akan bertambah banyak tetapi risiko kesalahan (*outliers*) meningkat. Jika diturunkan (misal ke 0.5), hasil match sangat akurat tetapi jumlahnya mungkin terlalu sedikit untuk membentuk model homografi.

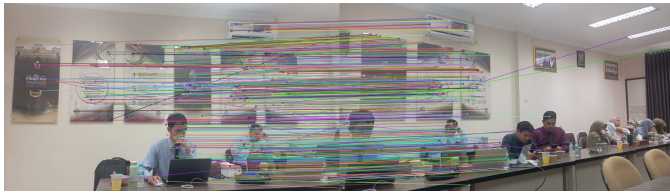


Fig. 3. Hasil Feature Matching setelah disaring dengan Lowe's Ratio Test.

D. Bagian D: Panorama Sederhana

Tahap akhir adalah penggabungan citra. Matriks transformasi (Homografi) dihitung menggunakan algoritma RANSAC untuk meminimalkan dampak data pencocokan yang salah.

Analisis Perluasan Kanvas: Perluasan kanvas mutlak diperlukan karena operasi *warping* mengubah geometri citra sumber (kanan) menyesuaikan perspektif citra tujuan (kiri). Tanpa perhitungan ukuran kanvas baru dan translasi (pergeseran) koordinat, bagian gambar hasil transformasi yang berada di koordinat negatif atau di luar dimensi asli akan terpotong (hilang). Implementasi yang dilakukan menghitung *bounding box* dari gabungan kedua citra untuk memastikan seluruh panorama terlihat utuh.



Fig. 4. Hasil akhir panorama (Stitching) dari dua citra.

III. REFLEKSI KONSEPTUAL (BAGIAN E)

A. Stabilitas SIFT vs Harris Corner

SIFT lebih stabil dibandingkan Harris Corner Detector karena SIFT memiliki sifat *Scale-Invariant* (tahan terhadap perubahan skala). Harris Corner bekerja sangat baik pada rotasi, namun akan gagal jika ukuran objek berubah (misalnya kamera menjauh) karena sudut (*corner*) mungkin tidak lagi terdeteksi sebagai sudut pada skala yang berbeda. SIFT mengatasi ini dengan mencari fitur ekstrem lokal pada ruang skala (DoG).

B. Kegagalan SIFT

Meskipun tangguh, SIFT dapat mengalami kegagalan pada kondisi:

- **Low Texture:** Citra yang sangat polos (misalnya langit biru cerah atau dinding putih) tidak memiliki gradien yang cukup untuk membentuk keypoint.
- **Repetitive Patterns:** Pola berulang (seperti pagar tralis yang identik) dapat membingungkan proses *matching* karena deskriptor satu titik sangat mirip dengan titik lainnya.
- **Motion Blur Ekstrem:** Kekaburan akibat gerakan cepat dapat merusak informasi gradien yang menjadi dasar deskriptor SIFT.

C. Relevansi SIFT di Era Deep Learning

SIFT masih relevan. Meskipun Deep Learning (seperti CNN atau Vision Transformers) mendominasi tugas klasifikasi dan deteksi objek kompleks, SIFT tetap unggul dalam skenario:

- **Data Sedikit:** SIFT tidak memerlukan proses *training* dengan ribuan data.
- **Efisiensi Sumber Daya:** SIFT dapat berjalan di perangkat dengan komputasi rendah (seperti drone atau mikrokontroler) tanpa memerlukan GPU mahal.
- **Akurasi Geometris:** Untuk tugas geometri presisi seperti *Structure from Motion* (SfM), SIFT seringkali memberikan lokalisasi fitur yang lebih akurat dibanding fitur *deep learning* generik.

IV. KESIMPULAN

Implementasi algoritma SIFT dalam tugas ini berhasil mendemonstrasikan kemampuan metode tersebut dalam mendeteksi fitur yang invarian dan melakukan penyambungan citra (*stitching*). Penggunaan DoG terbukti efisien untuk deteksi fitur multi-skala, dan penerapan RANSAC sangat krusial dalam menghasilkan matriks homografi yang akurat untuk panorama yang mulus.

REFERENCES

- [1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [2] OpenCV Documentation, "Feature Detection and Description," [Online]. Available: <https://docs.opencv.org/>.