

MENGGABUNGKAN BEBERAPA GAMBAR UNTUK MEMBENTUK PANORAMA MENGUNAKAN OPENCV PYTHON

Afrizal Zein

Dosen Tetap Program Studi Teknik Informatika, STMIK Eresha

e-mail : zeinafrizal@gmail.com, dono.yte@gmail.com

ABSTRAK

Foto-foto pemandangan alam yang sangat cantik, kita bisa gunakan ini untuk digunakan sebagai wallpaper komputer kita, akan tetapi tidak semua kamera atau hp bias mengambil gambar secara panorama, melainkan gambar tersebut terpotong atau tidak utuh.

Penelitian ini mencoba untuk mengambil dan menggabungkan beberapa foto sehingga menjadi utuh satu kesatuan. Metoda yang digunakan dengan memanfaatkan pustaka OpenCV Python membahas dasar-dasar algoritma penjahitan gambar yang khas, dan minimal, membutuhkan langkah-langkah utama:

1. Mendeteksi titik kunci (DoG, Harris, dll.) 2. Dan mengekstraksi deskriptor invarian lokal (SIFT, SURF, dll.).3. Dari dua gambar input, mencocokkan deskriptor di antara gambar, menggunakan algoritma RANSAC untuk memperkirakan matriks homografi menggunakan vektor fitur kami yang cocok. 4. Terakhir menerapkan transformasi warping menggunakan matriks homografi yang diperoleh dari Langkah sebelumnya namun, masalah terbesar dengan implementasi saya adalah bahwa mereka tidak mampu menangani lebih dari dua gambar input.

Dalam penelitian ini, kami akan meninjau kembali jahitan gambar dengan OpenCV, termasuk cara menjahit lebih dari dua gambar bersama menjadi gambar panorama.

Kata kunci : Penggabungan beberapa gambar, Panorama, Ransac, OpenCV dan Fitur Invarian

1. PENDAHULUAN

Pertama-tama mari kita memahami konsep mosaik atau menjahit gambar. Pada dasarnya jika Anda ingin mengabadikan adegan besar. Sekarang kamera Anda hanya dapat memberikan gambar dengan resolusi tertentu dan resolusi itu, katakanlah 640 x 480, tentu saja tidak cukup untuk menangkap tampilan panorama yang besar. Jadi, apa yang bisa dilakukan adalah menangkap beberapa gambar dari seluruh adegan dan kemudian menyatukan semua potongan-potongan menjadi satu tikar besar gambar. Ya, sepertinya bagus .. benar! Foto-foto seperti itu, yang berpose sebagai kumpulan adegan yang dipesan disebut sebagai mosaik atau panorama. Seluruh proses memperoleh beberapa gambar dan mengubahnya menjadi panorama seperti itu disebut sebagai mosaicking gambar. Dan akhirnya, kami memiliki satu foto besar pemandangan indah yang besar dan besar.

Metode lain untuk mencapai ini, adalah dengan menggunakan lensa sudut lebar di kamera Anda. Apa yang dilakukan lensa sudut lebar, secara efektif meningkatkan bidang pandang Anda. Outputnya, akan berbeda (jelas). Tetapi untuk keperluan tutorial ini, mari kita masuk ke cara membuat panorama menggunakan komputer dan bukan lensa.

Dalam makalah ini kami menjelaskan

pendekatan berbasis fitur invarian untuk menjahit gambar panorama sepenuhnya otomatis. Ini memiliki beberapa keunggulan dibandingkan pendekatan sebelumnya. Pertama, penggunaan kami terhadap fitur invarian memungkinkan pencocokan yang andal dari urutan gambar panorama meskipun rotasi, zoom dan iluminasi berubah pada gambar input. Kedua, dengan melihat jahitan gambar sebagai masalah pencocokan multi-gambar, kita bisa secara otomatis menemukan hubungan yang cocok antara gambar, dan mengenali panorama dalam dataset tidak terurut. Ketiga, kami menghasilkan hasil berkualitas tinggi menggunakan multi-band memadukan untuk membuat panorama output yang mulus. Penelitian ini memperluas pekerjaan kami sebelumnya di area [BL03] dengan memperkenalkan dan mendapatkan kompensasi dan langkah pelurusan otomatis. Kita juga menjelaskan implementasi penyesuaian bundel yang efisien dan menunjukkan bagaimana melakukan multi-band blending untuk beberapa gambar yang tumpang tindih dengan sejumlah band. Sisa makalah ini disusun sebagai berikut. Bagian 2 mengembangkan geometri masalah dan memotivasi pilihan kami dari fitur invarian. Bagian 3 menggambarkan gambar kita metodologi pencocokan (RANSAC) dan probabilitas model untuk verifikasi kecocokan

gambar. Di bagian 4 kami jelaskan algoritma penyesuaian gambar kami (penyesuaian bundel) yang bersama-sama mengoptimalkan parameter dari masing-masing kamera. Bagian 5 - 7 menggambarkan pipa render termasuk otomatis meluruskan, mendapatkan kompensasi dan multi-band pencampuran. Di bagian 9 kami menyajikan kesimpulan dan ide untuk pekerjaan masa depan.



Gambar 3.1 Pustaka OpenCV Python.

2. TINJAUAN PUSTAKA

Penelitian yang sudah pernah dilakukan yang berhubungan dengan penelitian ini adalah :

a. Matthew Brown and David G. Lowe (2016) Menurut Matthew Brown and David G. Lowe (dalam jurnalnya *Automatic Panoramic Image Stitching using Invariant Features*, 2016) Makalah ini membahas masalah sepenuhnya otomatis jahitan gambar panorama. Meskipun masalah 1D (tunggal sumbu rotasi) dipelajari dengan baik, jahitan 2D atau multi-baris lebih sulit. Pendekatan sebelumnya telah menggunakan input manusia atau pembatasan urutan gambar untuk menetapkan gambar yang cocok. Dalam karya ini, kami merumuskan jahitan sebagai masalah pencocokan multi-gambar, dan gunakan fitur lokal yang tidak berubah untuk menemukan kecocokan antara semua gambar. Karena ini metode kami tidak sensitif terhadap pemesanan, orientasi, skala dan iluminasi gambar input. Itu juga tidak sensitif untuk meredam gambar yang bukan bagian dari panorama, dan dapat mengenali beberapa panorama dalam gambar yang tidak terurut dataset. Selain memberikan lebih banyak detail, makalah ini meluas pekerjaan kami sebelumnya di area [BL03] dengan memperkenalkan dapatkan kompensasi dan langkah pelurusan otomatis.

b. Harrison Chau (2014)

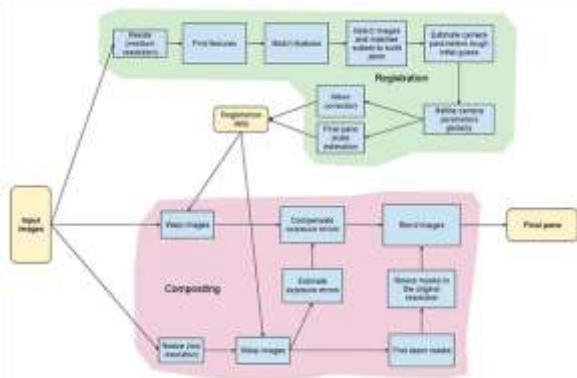
Dalam publikasi ilmiah berjudul "*Robust Panoramic Image Stitching*", menjabarkan penciptaan panorama menggunakan visi komputer bukanlah hal baru. Ide, bagaimanapun, sebagian besar algoritma berfokus pada menciptakan panorama menggunakan semua gambar dalam direktori. Metode yang akan digunakan dijelaskan secara rinci mengambil pendekatan ini selangkah lebih maju dengan tidak membutuhkan gambar di setiap panorama dipisahkan secara manual. Sebagai gantinya, ini mengelompokkan satu set gambar ke dalam panorama terpisah berdasarkan pencocokan fitur skala invarian. Kemudian gunakan cluster yang terpisah ini untuk menjahit gambar panorama.

3. METODA PENELITIAN

RANSAC (konsensus sampel acak) [FB81] adalah prosedur estimasi yang menggunakan set minimal acak korespondensi sampel untuk memperkirakan transformasi gambar parameter, dan menemukan solusi yang memiliki konsensus terbaik dengan data. Dalam hal panorama kami memilih set

$r = 4$ korespondensi fitur dan hitung homografinya H di antara mereka menggunakan transformasi linear langsung (DLT) metode [HZ04]. Kami ulangi ini dengan $n = 500$ percobaan RANSAC (konsensus sampel acak) [FB81] adalah kuat prosedur estimasi yang menggunakan set minimal acak korespondensi sampel untuk memperkirakan transformasi gambar parameter, dan menemukan solusi yang memiliki konsensus terbaik dengan data. Dalam hal panorama kami memilih set $r = 4$ korespondensi fitur dan hitung homografinya H di antara mereka menggunakan transformasi linear langsung (DLT) metode [HZ04]. Kami ulangi ini dengan $n = 500$ percobaan dan pilih solusi yang memiliki jumlah maksimum inliers (yang proyeksi konsisten dengan H dalam sebuah toleransi? piksel). Diberi probabilitas bahwa fitur kecocokan benar antara sepasang gambar yang cocok (yang sebelumnya probabilitas) adalah π , probabilitas untuk menemukan yang benar transformasi setelah n percobaan adalah p (H benar) $= 1 - (1 - \pi)^r n$. (6) Setelah sejumlah besar percobaan kemungkinan menemukan homografi yang benar sangat tinggi. Misalnya, untuk inlier probabilitas $\pi = 0,5$, probabilitas yang benar homografi tidak ditemukan setelah sekitar 500 percobaan 1×10^{-14} . RANSAC pada dasarnya adalah pendekatan pengambilan sampel untuk memperkirakan H . Jika bukannya memaksimalkan jumlah inliers satu memaksimalkan jumlah kemungkinan log, hasilnya maksimal estimasi kemungkinan (MLE). Selanjutnya, jika prior pada parameter transformasi yang tersedia, kita bias menghitung maksimum perkiraan posteriori (MAP). Ini algoritma adalah kn dan pilih solusi yang memiliki jumlah maksimum inliers (yang proyeksi konsisten dengan H dalam sebuah toleransi?)

piksel). Diberi probabilitas bahwa fitur kecocokan benar antara sepasang gambar yang cocok (yang sebelumnya probabilitas) adalah π , probabilitas untuk menemukan yang benar transformasi setelah n percobaan adalah $p(H \text{ benar}) = 1 - (1 - (\pi) r)^n$. (6) Setelah sejumlah besar percobaan kemungkinan menemukan homografi yang benar sangat tinggi. Misalnya, untuk inlier probabilitas $\pi = 0,5$, probabilitas yang benar homografi tidak ditemukan setelah sekitar 500 percobaan 1×10^{-14} . RANSAC pada dasarnya adalah pendekatan pengambilan sampel untuk memperkirakan H . Jika bukannya memaksimalkan jumlah inliers satu memaksimalkan jumlah kemungkinan log, hasilnya maksimal estimasi kemungkinan (MLE). Selanjutnya, jika prior pada parameter transformasi yang tersedia, kita bias menghitung maksimum perkiraan posteriori (MAP). Ini algoritma adalah kn



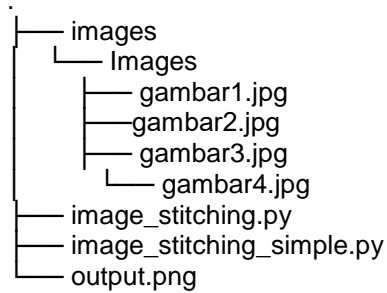
Gambar 3.2 Algoritma menggabungkan beberapa gambar

Algoritma yang akan kita gunakan di sini mirip dengan metode yang diusulkan oleh Brown dan Lowe dalam makalah 2017 mereka, Jahitan Gambar Panoramik Otomatis dengan Fitur Invarian. Tidak seperti algoritma penjahitan gambar sebelumnya yang sensitif terhadap urutan gambar input, metode Brown dan Lowe lebih kuat, membuatnya tidak peka terhadap: memesan gambar, orientasi gambar dan perubahan iluminasi. Gambar bising yang sebenarnya bukan bagian dari panorama Selain itu, metode jahitan gambar mereka mampu menghasilkan gambar panorama output yang lebih estetis melalui penggunaan kompensasi gain dan pencampuran gambar.

4. HASIL DAN PEMBAHASAN

Mari kita lihat bagaimana proyek ini diorganisasikan dengan perintah tree:

```
$ tree --dirsfirst
```



Gambar input masuk dalam gambar / folder. Saya memilih untuk membuat subfolder untuk scottsdale / set gambar saya seandainya saya ingin menambahkan subfolder tambahan di sini nanti.

```

# impor paket yang diperlukan
from imutils import paths
import numpy as np
import argparse
import imutils
import cv2
  
```

Paket yang diperlukan kami diimpor pada Baris 2-6. Khususnya, kita akan menggunakan OpenCV dan imutils. Jika Anda belum melakukannya atau belum ada maka, silakan download dan install:

```

# membangun parser argumen dan parsing argument
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--images", type=str, required=True, help="path to input directory of images to stitch")
ap.add_argument("-o", "--output", type=str, required=True, help="path to the output image")
args = vars(ap.parse_args())
# ambil gambar input dan inialisasi daftar gambar
print("[INFO] loading images...")
imagePaths = sorted(list(paths.list_images(args["images"])))
images = []

# loop di atas jalur gambar, muat masing-masing gambar dan tambahkan ke gambar untuk dijahit
for imagePath in imagePaths:
    image = cv2.imread(imagePath)
    images.append(image)
  
```

Kemudian untuk setiap imagePath, kami akan memuat gambar dan menambahkannya ke daftar gambar (Baris 19-25).

Sekarang setelah gambar berada dalam memori, mari kita lanjutkan dan menjahitnya menjadi satu panorama menggunakan kemampuan bawaan OpenCV:



Gambar 4.1 Potongan gambar pertama



Gambar 4.2 Potongan gambar kedua



Gambar 4.3 Potongan gambar ketiga



Gambar 4.4 Potongan gambar keempat



Gambar 4.5 Hasil penggabungan gambar yang membentuk panorama

5. KESIMPULAN

Kesimpulan yang diperoleh pada penelitian ini adalah :

1. Dengan menggunakan OpenCV dan Python, kami dapat menyatukan beberapa gambar sekaligus dan membuat gambar panorama.
2. Gambar panorama keluaran kami tidak hanya akurat dalam penempatan jahitannya tetapi juga menyenangkan secara estetika.
3. Namun, salah satu kelemahan terbesar dari menggunakan kelas menjahit gambar bawaan OpenCV adalah bahwa hal itu mengabstraksi banyak komputasi internal, termasuk matriks homografi yang dihasilkan sendiri.

SARAN

1. Jika Anda mencoba melakukan penjahitan gambar waktu-nyata, Anda mungkin merasa terbantu dengan men-cache matriks homografi dan hanya sesekali melakukan deteksi keypoint, ekstraksi fitur, dan pencocokan fitur.
2. Melewati langkah-langkah ini dan menggunakan matriks cache untuk melakukan warping perspektif dapat mengurangi beban komputasi pipa Anda dan pada akhirnya mempercepat algoritma penjahitan gambar waktu-nyata, tetapi

sayangnya, `cv2.createStitcher` Python OpenCV tidak memberi kami akses ke matriks mentah.

6. DAFTAR PUSTAKA

- [1] Matthew Brown and David G. Lowe(2016), ***Automatic Panoramic Image Stitching using Invariant Features***, Department of Computer Science,University of British Columbia, Vancouver, Canada, 2016
- [2] Richard Szeliski (2006), ***Image Alignment and Stitcing***, Computer Graphics and Vision Vol. 2, No 1 (2006) 1–104 c 2006
- [3] Alba Pujol Miró, ***Real-Time Image Stitching For Automotive 360°Vision System***, Barcelona, July 2014
- [4] Harrison Chau (2015), ***Robust Panoramic Image Stitching***. Department of Aeronautics and Astronautics Stanford University Stanford, CA, USA,2014
- [5] Afrizal Zein (2016), ***Pendeteksian Multi Wajah Dan Recognition Secara Real Time Menggunakan Metoda Principal Component Analysis (Pca) Dan Eigenface***, Jurnal ESIT STMIK ERESHA ,2016
- [6] Afrizal Zein (2018), ***Menggabungkan Dua Wajah Dengan Metoda Ensemble Regression Trees Menggunakan Pustaka Dlib Dan Opencv Python***, Jurnal ESIT STMIK ERESHA ,2018.
- [7] Afrizal Zein (2018), ***Peran Text Processing Dalam Aplikasi Penerjemah Multi Bahasa Menggunakan Ajax Api Google***, Jurnal ESIT STMIK ERESHA ,2018.
- [8] Afrizal Zein (2018),, ***Pendeteksian Kantuk Secara Real Time Menggunakan Pustaka Opencv Dan Dlib Python***, Sainstech: Jurnal Penelitian dan Pengkajian Sains, 2018