

Rekonstruksi dan Visualisasi Objek 3-D Berbasis Algoritma *Direct Linear Transformation*

Sunu Wibirama¹, Risanuri Hidayat²

Abstract— Visual reconstruction and modeling of three dimensional (3-D) object is an emerging research topic in computer vision and photogrammetry fields. Nowadays, many commercial products are developed to obtain 3-D model of rigid object, starting from small ancient heritages to city landscape. Such commercial tools are very expensive and not accessible for education and research purpose. This paper aims to present a low-cost approach to generate a 3-D model from geometry of multiple two-dimensional (2-D) images using consumer-level Digital Single Lens Reflection (DSLR) camera. Direct Linear Transformation (DLT) algorithm was used to obtain 3-D point cloud. Texturization of 3-D object was generated by implementing Convex Hull and Random Sample Consensus (RANSAC) algorithms. Experimental result of small Merlion Singapore statue and large Herz-Jesu building shows that the proposed low-cost method is able to visually reconstruct small and large objects by using 10-20% feature points detected on 2D images.

Intisari— Rekonstruksi visual dan pemodelan objek tiga dimensi (3-D) adalah sebuah topik penelitian yang sedang berkembang pesat dalam bidang *computer vision* dan fotogrametri. Saat ini, banyak produk komersial dikembangkan untuk melakukan pemodelan objek 3-D dengan struktur *rigid*, mulai dari warisan sejarah berukuran kecil sampai dengan bentang darat perkotaan. Alat-alat komersial tersebut biasanya berbiaya tinggi dan tidak terjangkau apabila digunakan untuk tujuan pendidikan dan penelitian. Paper ini memperkenalkan sebuah metode berbiaya rendah untuk melakukan proses rekonstruksi objek 3-D dari unsur geometri beberapa citra dua dimensi (2-D) dengan menggunakan kamera *Digital Single Lens Reflection* (DSLR) yang banyak dijumpai di pasaran. Algoritma *Direct Linear Transformation* (DLT) digunakan untuk menghasilkan titik-titik 3-D. Proses teksturisasi objek 3-D dilakukan dengan memanfaatkan algoritma *Convex Hull* dan *Random Sample Consensus* (RANSAC). Hasil percobaan pada objek patung miniatur Merlion Singapore dan bangunan Herz-Jesu menunjukkan bahwa metode berbiaya rendah ini mampu merekonstruksi objek miniatur dan objek berukuran besar secara visual dengan memanfaatkan 10-20% titik-titik fitur yang terdeteksi pada citra 2-D.

Kata Kunci— Rekonstruksi visual, komputasi fotografi, *Direct Linear Transformation*, RANSAC, *Convex Hull*

I. PENDAHULUAN

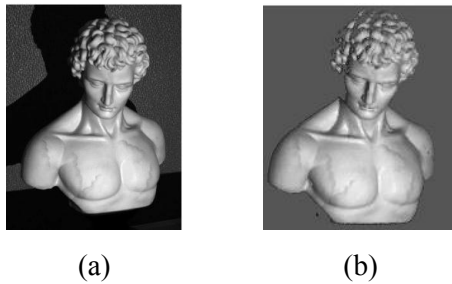
Dalam beberapa tahun terakhir, teknologi grafika komputer (*computer graphics*) telah mengalami kemajuan yang amat

pesat dalam memvisualisasikan model 3-D. Berbagai macam teknik telah mencapai tahap kematangan dan diimplementasikan dalam perangkat keras grafis. Apabila dalam beberapa tahun yang lalu proses visualisasi membutuhkan biaya yang tidak sedikit dan hanya komputer dengan biaya tinggi saja yang mampu melakukan proses visualisasi 3-D, saat ini komputer personal atau laptop dengan harga yang terjangkau bahkan digunakan untuk memvisualisasikan model 3-D yang dapat digerakkan secara interaktif, seperti yang terdapat pada *console game* maupun visualisasi data saintifik secara *real-time*, seperti gerakan magma [1], kandungan panas bumi [2], dan data medis [3, 4].

Perubahan ini membawa dampak yang signifikan terhadap kemungkinan tingkat kerumitan objek 3-D yang divisualisasikan. Permasalahan yang muncul adalah, secanggih apapun teknologi visualisasi dan kemampuan komputasi perangkat keras yang dimiliki, proses sintesis model 3-D yang terlihat realistis (memiliki tingkat detail yang hampir setara atau setara dengan objek yang sesungguhnya) bukanlah hal yang mudah dan membutuhkan waktu cukup lama, sehingga membutuhkan usaha yang memakan biaya. Berbagai macam model 3-D terinspirasi oleh objek-objek riil yang ada dalam kehidupan sehari-hari. Akan sangat baik apabila model 3-D tersebut bisa disintesis dengan cara langsung mengambil bagian-bagian dari objek riil secara visual. Gbr. 1 menunjukkan perbandingan antara objek riil patung (Gbr. 1a) dan hasil rekonstruksinya secara visual (Gbr. 1b) dengan menggunakan 3-D *scanner* komersial. Dari hasil rekonstruksi tersebut, dapat kita lihat tingkat akurasi model 3-D yang dihasilkan mendekati objek riil yang direkonstruksi.

Para peneliti telah melakukan berbagai macam upaya untuk melakukan rekonstruksi objek 3-D secara visual dari citra dua dimensi (2-D). Pada masa lampau, proses pengenalan dan rekonstruksi objek 3-D ini banyak digunakan pada aplikasi robotika [5-7]. Saat ini, kebutuhan untuk memvisualisasikan objek 3-D banyak digunakan dalam aplikasi animasi dan grafis [8], arsitektur [9], pendidikan dan pengenalan budaya [10], maupun *virtual reality* [11]. Hal ini membawa perubahan pada penekanan proses sintesa model 3-D tersebut. Kualitas kerumitan secara visual menjadi hal penting yang perlu mendapatkan porsi perhatian yang besar. Tidak hanya posisi beberapa titik 3-D saja yang perlu diukur secara akurat, kualitas tampilan dan aspek geometris dari seluruh titik yang terdapat pada suatu permukaan objek 3-D juga perlu diukur dengan teliti sehingga menambah tingkat kerumitan proses sintesa model 3-D dari objek riil.

^{1, 2} Dosen Jurusan Teknik Elektro dan Teknologi Informasi
Fakultas Teknik Universitas Gadjah Mada, Jln. Grafika 2
Yogyakarta 55281 INDONESIA (telp: 0274-5555; fax: 0274-4321;
e-mail: sunu@jtetl.gadjahmada.edu)



Gbr. 1 Rekonstruksi objek 3-D secara visual: (a) Objek riil berupa patung; (b) Hasil rekonstruksi visual menggunakan 3D scanner komersial. (Courtesy : Douglas Lanman – Massachusetts Institute of Technology)

Proses penggalian data visual dan kerumitan teknis ini seringkali tidak bisa terpenuhi dengan kondisi perangkat keras yang ada, sehingga memerlukan proses kalibrasi setiap kali peralatan tersebut digunakan untuk memperoleh informasi visual dari citra 2-D. Hal ini menjadi kendala tersendiri pada proses penggalian data visual. Seharusnya, proses kalibrasi alat ditekan seminimal mungkin, sehingga proses penggalian data visual menjadi lebih singkat. Di sisi lain, sistem 3D-scanner yang sudah ada seringkali dibangun dengan perangkat keras komersial khusus, seperti *laser range finders* atau LIDAR (*Light Detecting and Ranging*), RADAR (*Radio Detecting and Ranging*) maupun *Stereo Rig* komersial, sehingga memerlukan biaya yang tidak sedikit dan tidak terjangkau apabila digunakan untuk sarana edukasi dan penelitian (> Rp. 20 Juta) [12]. Selain itu, peralatan komersial seringkali merahasiakan teknik dan algoritma yang digunakan (*proprietary algorithm*), sehingga penelitian dengan alat komersial terbatas pada algoritma yang digunakan oleh peralatan tersebut.

Pada paper ini, kami memperkenalkan sebuah teknik rekonstruksi visual berbiaya rendah dengan teknik komputasi fotografi berbasis algoritma *Direct Linear Transformation* (DLT). Penelitian ini menitikberatkan pada penggunaan perangkat keras dengan biaya yang terjangkau (< Rp. 5 Juta) dan pengembangan algoritma rekonstruksi untuk objek 3-D dengan struktur rigid, baik objek bangunan berukuran besar maupun objek miniatur berukuran kecil.

II. METODE PENELITIAN

A. Garis Besar Alur Rekonstruksi 3-D

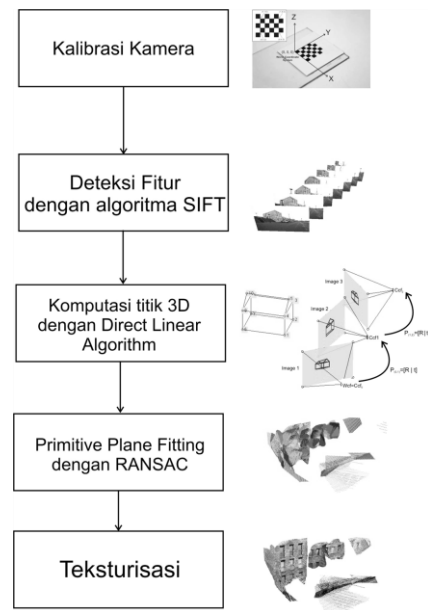
Penelitian di bidang rekonstruksi objek 3-D dan pemodelan berbasis citra 2-D sering dikategorikan pada lingkup 3-D vision di bidang *computer vision*. 3-D vision secara garis besar membahas bagaimana manusia mempersepsikan objek 3-D pada sistem pengindraannya. Dengan memahami cara kerja penginderaan manusia, algoritma komputasi fotografi dapat dikembangkan [12]. Secara garis besar, bagan alir proses rekonstruksi objek 3-D pada penelitian ini dijelaskan pada Gbr. 2.

Sebelum melakukan proses rekonstruksi 3-D, pertama kali proses kalibrasi kamera harus dilakukan. Proses kalibrasi ini bertujuan untuk mendapatkan parameter kamera. Kamera yang terkalibrasi ini kemudian digunakan untuk mengambil citra 2-D dari objek 3-D yang akan direkonstruksi. Selanjutnya, deteksi titik-titik fitur secara otomatis dan masif pada citra-citra 2-D dilakukan dengan memanfaatkan

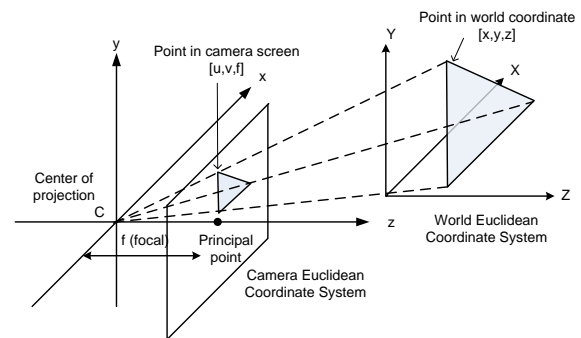
algoritma *Scale Invariant Feature Tracking* (SIFT) [15]. Koordinat-koordinat titik 2-D yang ditemukan disimpan dan digunakan sebagai input untuk algoritma *Direct Linear Transformation* (DLT) [20-22]. Algoritma DLT digunakan untuk menghasilkan koordinat 3-D dari koordinat 2-D dan parameter kamera. Selanjutnya, proses *primitive plane fitting* dengan memanfaatkan algoritma *Convex Hull* [24] dan RANSAC [25] dilakukan untuk menghasilkan bidang-bidang primitif yang siap diberi tekstur. Terakhir, proses teksturisasi dilakukan untuk memastikan model 3-D yang dihasilkan menyerupai objek riil yang direkonstruksi.

B. Model Kamera Pinhole dan Kalibrasi Kamera

Sebagian besar citra 2-D adalah hasil proyeksi adegan 3-D yang dihasilkan oleh sistem kamera. Proses proyeksi adegan 3-D ke citra 2-D ini didapatkan dengan mengolah persamaan transformasi perspektif (*perspective transformation*). Pada penelitian kali ini, model kamera lubang jarum (*pinhole*) digunakan untuk menangani persamaan transformasi perspektif tersebut [23]. Sebuah titik yang terletak di ruang 3-D bisa diekspresikan sebagai titik $[x \ y \ z \ 1]^T$ yang terletak di sistem koordinat dunia (*world coordinate system*). Titik $[x \ y \ z \ 1]^T$ tersebut dipotret dengan kamera dan mengalami transformasi linier dari ruang 3-D ke ruang proyeksi 2-D sebagaimana diilustrasikan oleh Gbr. 3.



Gbr. 2 Bagan alir dari proses rekonstruksi objek 3-D



Gbr. 3 Model kamera lubang jarum dengan transformasi perspektif

Titik koordinat homogen 2-D $[u \ v \ w]^T$ pada sistem koordinat kamera (*camera coordinate system*) diperoleh dengan menyelesaikan (1) di bawah ini:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = K [I_3 | O_3] \begin{bmatrix} R & -T \\ O_3^T & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (1)$$

atau

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = M \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \text{ dimana } M = [KR \ -KRT] \quad (2)$$

Matriks K dan $[I_3 | O_3]$ pada (1) adalah parameter intrinsik yang mengandung variabel-variabel fisik pada kamera, seperti fokus dan titik utama (*principal point*). Matriks K adalah matriks kamera kalibrasi yang didefinisikan sebagai berikut:

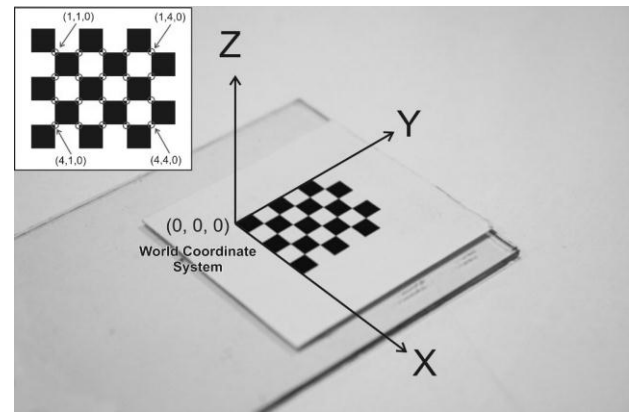
$$K = \begin{bmatrix} fa & 0 & u_0 \\ 0 & fb & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

di mana f adalah panjang fokus pada proyeksi kamera, a dan b adalah faktor konversi ukuran citra dari unit fisik (misal: cm atau mm) ke unit pixel pada sumbu x dan y , (u_0, v_0) adalah titik utama pada proyeksi kamera. Matriks $[I_3 | O_3]$ didefinisikan sebagai $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$. Matriks $\begin{bmatrix} R & -T \\ O_3^T & 1 \end{bmatrix}$ mengandung

parameter ekstrinsik dari kamera yang digunakan pada penelitian.

Parameter ekstrinsik mengandung informasi lokasi dan orientasi kamera di sistem koordinat dunia. Matriks rotasi R mengandung tiga elemen dasar rotasi sistem koordinat kamera terhadap sistem koordinat dunia. Vektor translasi T mengandung informasi perpindahan sistem koordinat kamera terhadap sistem koordinat dunia. Matriks O_3^T adalah vektor $[0 \ 0 \ 0]$. Matriks K dan $[I_3 | O_3]$ pada (1) adalah parameter intrinsik yang mengandung variabel-variabel fisik pada kamera.

Untuk memperoleh Matriks M pada (1), proses kalibrasi kamera dengan sebuah objek 3-D yang telah diketahui koordinat 3-D-nya perlu dilakukan. Pada penelitian kali ini, pola hitam putih papan catur digunakan pada proses kalibrasi kamera, sebagaimana ditunjukkan oleh Gbr. 4.



Gbr. 4 Pola hitam putih pada papan catur yang akan kami gunakan untuk melakukan proses kalibrasi kamera. Pada pola ini, kami mengasumsikan titik asal sistem koordinat dunia di bagian kiri atas dengan koordinat $z = 0$ untuk titik-titik yang akan kami deteksi. Ukuran fisik pola bujur sangkar harus diukur terlebih dahulu sebelum proses kalibrasi kamera dilakukan.

Apabila diasumsikan koordinat 3-D titik-titik papan catur adalah $(X=[x \ y \ z \ 1]^T)$, dan koordinat 2-D $[u \ v \ w]^T$ titik-titik tersebut didapatkan dengan algoritma deteksi fitur SIFT, maka akan didapatkan persamaan di bawah ini:

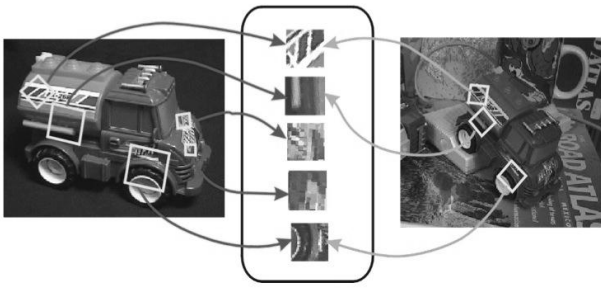
$$\begin{bmatrix} x & y & z & 1 & 0 & 0 & 0 & 0 & -ux & -uy & -uz & 1 \\ 0 & 0 & 0 & 0 & x & y & z & 1 & -vx & -vy & -vz & 1 \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ \vdots \\ m_{34} \end{bmatrix} = 0 \quad (4)$$

atau $AM = 0$

Untuk titik-titik pada papan catur sejumlah n , matriks A akan berukuran $2n \times 12$. Untuk mendapatkan matriks M , kami menggunakan teknik *Singular Value Decomposition* (SVD) dan menghasilkan formula $A = UDV^T$ [13]. Kolom terakhir matriks V adalah solusi dari matriks M . Dari matriks M tersebut, parameter intrinsik dan parameter ekstrinsik didapatkan dengan teknik *QR-decomposition* [14].

C. Deteksi Fitur dengan SIFT

Scale-Invariant Feature Transform (SIFT) adalah sebuah algoritma *computer vision* yang digunakan untuk mendeteksi dan mendeskripsikan titik-titik fitur lokal pada citra. Algoritma ini dikembangkan oleh David Lowe pada tahun 1999 [15]. Secara garis besar, algoritma ini menyediakan sebuah metode untuk mendeteksi titik-titik fitur pada sebuah citra 2-D yang dianggap memiliki karakter *invariant* (kuat dan tidak terpengaruh) terhadap transformasi geometris (*scaling, rotation, translation, noise*, dan perubahan kadar pencahayaan). Titik-titik ini yang memiliki karakter khusus ini sangat bermanfaat untuk mendeteksi sebuah objek yang terekam dalam berbagai posisi. Gbr. 5 menunjukkan bagaimana SIFT mampu mendeteksi titik-titik fitur pada sebuah objek, kemudian menyimpan data fitur tersebut, dan menggunakannya untuk mendeteksi objek yang sama dengan posisi pengambilan gambar yang berbeda.



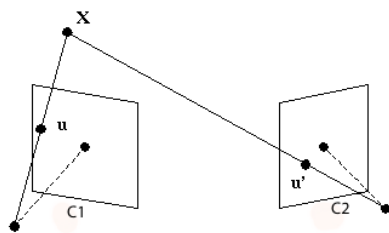
Gbr. 5 Deteksi titik-titik fitur dengan algoritma SIFT. Algoritma SIFT mampu mendeteksi titik-titik fitur yang *invariant* terhadap transformasi geometris, seperti *scaling*, rotasi, dan translasi.

Selain itu, karakteristik lain yang sangat penting pada algoritma SIFT adalah posisi relatif dari fitur-fitur yang terdeteksi pada sebuah objek tidak boleh berubah antara satu citra dengan citra yang lain. Sebagai contoh, apabila untuk mendeteksi objek jendela digunakan empat titik sudut jendela, maka algoritma SIFT akan bekerja dengan baik meskipun objek jendela direkam dalam posisi yang berbeda. Namun demikian, apabila titik fitur pada daun jendela dilibatkan dalam proses tersebut, ada kemungkinan SIFT akan gagal bekerja mendeteksi objek jendela dalam citra lain apabila pada citra tersebut daun jendela dibuka atau ditutup.

SIFT bekerja dengan menerapkan operasi *filtering* bertingkat, dimana operasi *filtering* yang memiliki kompleksitas lebih tinggi akan diterapkan pada titik-titik fitur yang telah diseleksi oleh serangkaian pengujian awal.

D. Rekonstruksi Koordinat 3-D dengan DLT

Apabila parameter kamera telah diketahui, proses rekonstruksi titik 3-D dari titik-titik 2-D pada citra bisa dilakukan dengan teknik *Direct Linear Transformation* (DLT) [20-22]. Untuk melakukan proses rekonstruksi, sekurang-kurangnya dua citra objek yang diambil dari posisi yang berbeda dibutuhkan, sebagaimana diilustrasikan pada Gbr. 6.



Gbr. 6 Proses rekonstruksi satu titik 3-D dari dua titik 2-D

Pada proses kalibrasi kamera, matriks *M* dihasilkan. Untuk dua kamera, matriks *M* untuk masing-masing kamera akan dilambangkan sebagai *M'* dan *M''*. Apabila (*u',v', w'*) dan (*u'', v'', w''*) adalah titik-titik 2-D yang kami hasilkan pada dua buah citra yang dipotret kamera, maka kami bisa melakukan rekonstruksi titik 3-D homogen ($X = [x \ y \ z]^T$) dengan menyelesaikan persamaan berikut:

$$\begin{bmatrix} u' m_3'^T - w' m_1'^T \\ v' m_3'^T - w' m_2'^T \\ u'' m_3''^T - w'' m_1''^T \\ v'' m_3''^T - w'' m_2''^T \end{bmatrix} X = AX = 0 \tag{5}$$

dimana m_i^T adalah baris ke-*i* dari matriks *M*, sedangkan *X* adalah matriks $[x \ y \ z \ 1]^T$. Dalam implementasinya, titik-titik yang sudah direkonstruksi ini nantinya akan digabungkan menjadi sebuah bidang primitif dan proses teksturisasi model 3-D akan dilakukan pada bidang primitif ini.

E. Teksturisasi dengan Primitive Plane Fitting

Setelah titik-titik 3-D dihasilkan, tahap berikutnya adalah mengaplikasikan *primitive plane* (bidang-bidang primitif, seperti persegi atau persegi panjang) pada titik-titik 3-D tersebut. Bidang-bidang primitif ini nantinya menjadi dasar untuk proses teksturisasi dengan memanfaatkan algoritma *Convex Hull* [24]. Algoritma *Random Sample Consensus* (RANSAC) digunakan untuk optimisasi proses *primitive plane fitting* [25].

Untuk menghasilkan bidang-bidang primitif dari titik-titik 3-D, pertama kali beberapa titik diambil sebagai sampel dengan *m* titik terdekat (*consensus point*). Dengan menggunakan RANSAC, bidang primitif dihasilkan dan *outliers* dihilangkan. Jika sebuah bidang primitif ditemukan, maka titik-titik 3-D yang dikategorikan sebagai *outliers* tidak akan digunakan untuk proses *sampling* (produksi bidang primitif) berikutnya. Setelah satu bidang primitif ditemukan, secara otomatis akan dilakukan *sampling* pada beberapa titik lain, kemudian proses produksi bidang primitif dari titik-titik 3-D tersebut akan kembali dilakukan. Demikian seterusnya sampai seluruh titik terproses sebagai bidang primitif.

Teksturisasi dilakukan dengan mengimplementasikan tekstur objek riil. Untuk keperluan hal tersebut, algoritma *Convex Hull* digunakan untuk menentukan batas masing-masing bidang primitif. Untuk menghasilkan tekstur pada model 3-D, kami memproyeksikan tekstur dari citra objek riil yang bersesuaian dengan masing-masing bidang planar pada model 3-D.

F. Implementasi Metode Komputasi Fotografi

Penelitian kali ini menggunakan citra objek yang diambil secara manual oleh peneliti dan dataset yang diperoleh dari internet. Objek yang digunakan oleh peneliti adalah sebuah miniatur patung Merlion dengan dimensi 10 cm x 5 cm x 13 cm, sebagaimana ditampilkan pada Gbr. 7. Objek bangunan diambil dari dataset bangunan Herz-Jesu versi P-25 yang disediakan oleh Christoph Strecha [16,17]. Penggunaan dataset dari hasil penelitian yang sudah ada ini dimaksudkan agar hasil rekonstruksi dengan algoritma yang kami kembangkan dapat dibandingkan secara visual dengan algoritma yang sudah dikembangkan oleh peneliti sebelumnya [17]. Bangunan Herz-Jesu ditunjukkan pada Gbr. 8.

Pada penelitian ini, kami menggunakan komputer dengan prosesor Intel Core 2 Quad 2.66 GHz, memori 2 GB, dan kartu grafis NVIDIA GeForce 9500 GT. Perangkat lunak yang kami gunakan dalam penelitian ini meliputi MATLAB 2011, GML Calibration Toolbox [18], dan OpenSceneGraph 2.8.3 [19]. Pengambilan citra objek Merlion dilakukan dengan menggunakan kamera DSLR seri 1000D dengan lensa Canon EF-S 18-55 f/3.5-5.6.

Sebelum kamera digunakan untuk mengambil data citra Merlion, terlebih dahulu dilakukan kalibrasi kamera dengan menggunakan pola papan catur berdimensi 9 x 6. Jarak antara

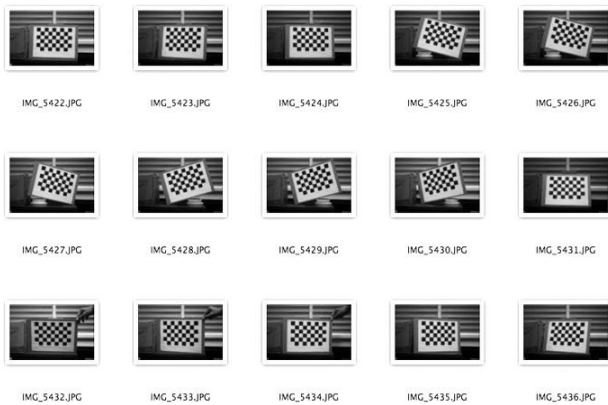
papan catur dan kamera sebesar 90 cm. Setiap persegi hitam dan putih pada pola papan catur memiliki ukuran sisi sebesar 28 cm. Pada penelitian ini, diambil citra pola catur pada 15 posisi yang berbeda sebagaimana ditunjukkan pada Gbr. 9



Gbr. 7 Miniatur patung Merlion yang digunakan dalam penelitian



Gbr. 8 Bangunan Herz-Jesu yang digunakan dalam penelitian

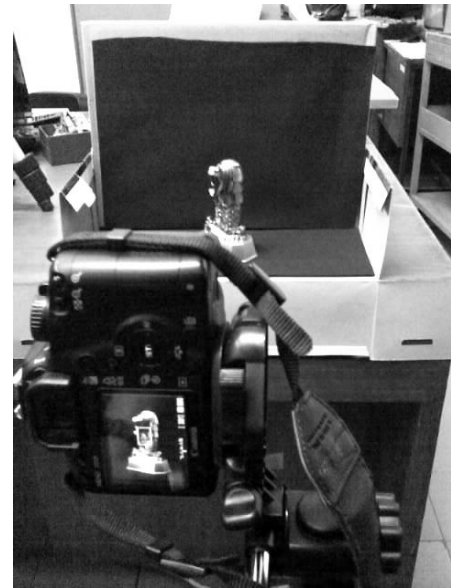


Gbr. 9 Lima belas posisi papan catur untuk kalibrasi kamera

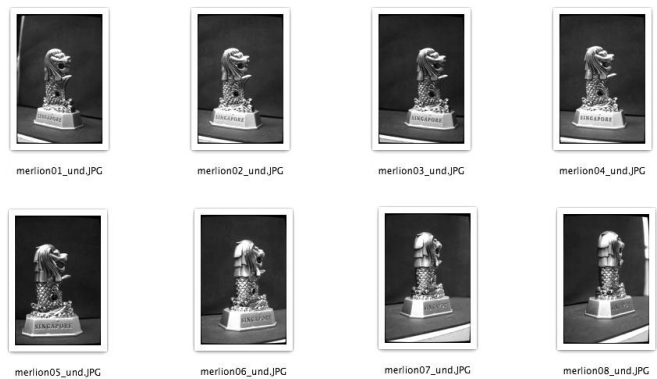
Setelah citra papan catur didapatkan, lima belas citra tersebut diolah dengan perangkat lunak GML Calibration Toolbox [18]. Instalasi perangkat keras untuk pengambilan

sampel data Merlion ditunjukkan oleh Gbr. 10. Jarak antara kamera DSLR 1000D dan objek kurang lebih 90 cm. Objek diletakkan di sebuah kotak kecil dengan latar belakang kain hitam. Penggunaan latar belakang hitam ini dimaksudkan untuk menyederhanakan kerja algoritma SIFT.

Tahap berikutnya adalah pengambilan delapan citra Merlion dari berbagai posisi. Citra Merlion dari berbagai posisi ini nantinya digunakan untuk membuat model 3-D dengan algoritma rekonstruksi objek tiga dimensi yang telah dijelaskan pada landasan teori. Citra Merlion yang digunakan pada penelitian ini memiliki resolusi 1152 x 1728 dengan tipe JPEG. Hasil pengambilan data Merlion dengan kamera DSLR dapat dilihat pada Gbr. 11.



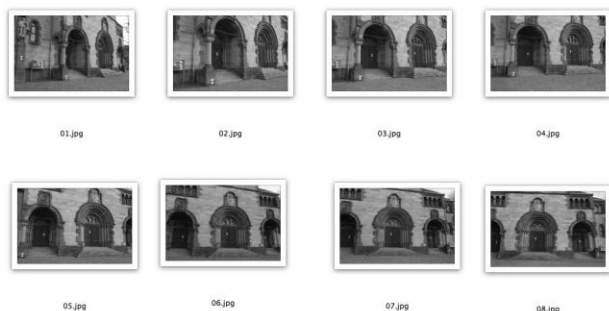
Gbr. 10 Pengambilan data citra Merlion dengan kamera DSLR 1000D



Gbr. 11 Citra Merlion dari berbagai posisi

Selain citra Merlion yang didapatkan secara manual, kami juga menggunakan citra bangunan Herz-Jesu dari dataset yang tersedia di *internet*. Sebagaimana percobaan sebelumnya, kami menggunakan delapan citra Herz-Jesu untuk melakukan proses rekonstruksi dengan resolusi masing-masing citra sebesar 3072 x 2048 dengan tipe JPEG. Citra diambil dengan menggunakan kamera DSLR Canon D60 [17]. Data parameter kamera bisa didapatkan pada [16]. Delapan citra Herz-Jesu ditunjukkan pada Gbr. 12. Untuk melakukan perbandingan

secara visual atas algoritma yang digunakan pada penelitian ini, kami mengunduh hasil rekonstruksi bangunan Herz-Jesu yang didapatkan dengan sistem multi-view stereo pada [17]. Citra yang akan digunakan sebagai pembandingan ditunjukkan pada Gbr. 13.



Gbr. 12 Citra Herz-Jesu dari berbagai posisi



Gbr. 13 Citra Herz-Jesu yang digunakan sebagai pembandingan

III. HASIL PERCOBAAN DAN DISKUSI

A. Hasil Rekonstruksi Bangunan Herz-Jesu

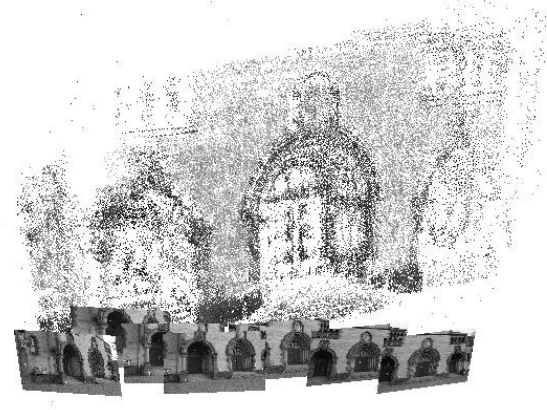
Untuk menampilkan model 3-D hasil rekonstruksi, kami menggunakan perangkat lunak OpenSceneGraph versi 2.3.8 yang bisa didapatkan tak berbayar pada website resmi OpenSceneGraph [19]. OSG merupakan pustaka open source yang digunakan untuk penelitian di bidang pemodelan 3-D atau visualisasi.

Dari titik-titik fitur yang terdeteksi menggunakan algoritma SIFT, dihasilkan titik-titik 3-D sebanyak 6241 titik yang sudah terekonstruksi dalam bidang-bidang primitif. Dengan demikian, hanya sekitar 17,36 % titik-titik fitur dari masing-masing citra yang digunakan untuk proses rekonstruksi 3-D objek Herz-Jesu.

Gbr. 14 menunjukkan hasil rekonstruksi titik-titik 3-D dan citra bangunan Herz-Jesu yang digunakan dalam penelitian ini. Hanya titik-titik 3-D yang dikategorikan sebagai *inliers* saja yang nantinya akan digunakan pada proses *fitting* bidang-bidang primitif. Gbr.15 menunjukkan hasil akhir proses teksturisasi setelah proses *fitting* bidang-bidang primitif dilakukan. Untuk menghasilkan model 3-D sebagaimana Gbr.15, kami menggunakan algoritma RANSAC dengan 500 kali iterasi.

Gbr. 16 menunjukkan perbandingan pengaruh jumlah iterasi dan model 3-D yang dihasilkan. Gbr. 16.a dan Gbr. 16.b adalah model 3-D bangunan Herz-Jesu yang dihasilkan dengan algoritma RANSAC 200 kali iterasi. Gbr 16.c dan Gbr. 16.d adalah model 3-D bangunan Herz-Jesu yang dihasilkan dengan algoritma RANSAC 700 kali iterasi. Dari Gbr. 16, secara visual dapat dilihat bahwa jumlah iterasi RANSAC berpengaruh pada hasil akhir model 3-D. Semakin besar iterasi, jumlah bidang-bidang primitif yang ditemukan akan semakin banyak dan pada akhirnya akan meningkatkan kualitas tekstur dari model 3-D yang dihasilkan.

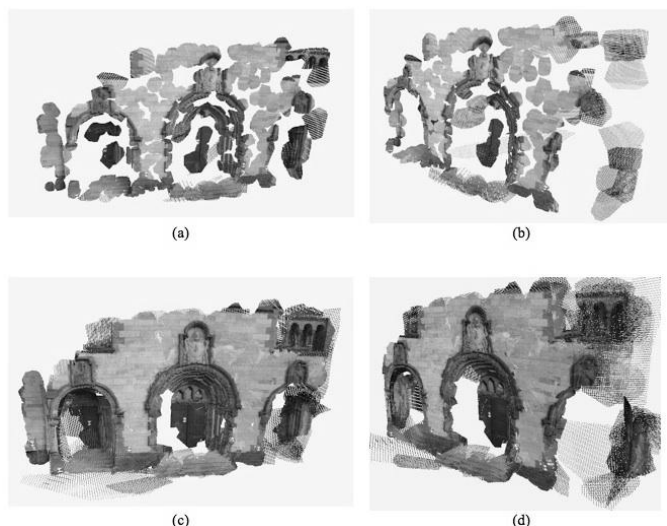
Gbr. 17 menunjukkan perbandingan antara hasil rekonstruksi yang dilakukan oleh Christoph Strecha [17] dan hasil rekonstruksi dengan teknik komputasi fotografi kami. Secara garis besar, teknik komputasi fotografi mampu menghasilkan model 3-D yang menyerupai objek riil yang direkonstruksi. Perbedaan utama antara hasil rekonstruksi pada Gbr. 17.a dan Gambar 17.b adalah pada bentuk tekstur yang dihasilkan. Gbr. 17.a memiliki tekstur yang sedikit lebih kasar dibandingkan dengan Gbr. 17.b. Namun demikian, Gbr. 17.a mampu menampilkan detail tekstur yang lebih terperinci apabila dibandingkan dengan Gbr. 17.b. Untuk menghasilkan tekstur model yang lebih baik, diperlukan penelitian dan pengembangan lebih lanjut dari algoritma *primitive plane fitting* atau rekonstruksi permukaan 3-D. Dengan memanfaatkan algoritma rekonstruksi permukaan yang tepat untuk objek dengan karakteristik tekstur yang berbeda, diharapkan model 3-D yang dihasilkan akan semakin baik.



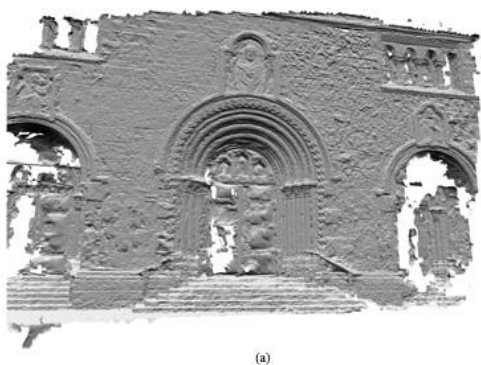
Gbr. 14 Hasil rekonstruksi titik-titik 3-D dengan algoritma DLT



Gbr. 15 Hasil proses teksturisasi titik-titik 3-D menggunakan algoritma RANSAC dengan 500 kali proses iterasi.



Gbr. 16 Rekonstruksi objek bangunan Herz-Jesu: (a) Rekonstruksi dengan iterasi RANSAC 200; (b) Rekonstruksi dengan iterasi RANSAC 200, dilihat dari sudut 45° searah jarum jam; (c) Rekonstruksi dengan iterasi RANSAC 700; (d) Rekonstruksi dengan iterasi RANSAC 700, dilihat dari sudut 45° searah jarum jam.



(a)



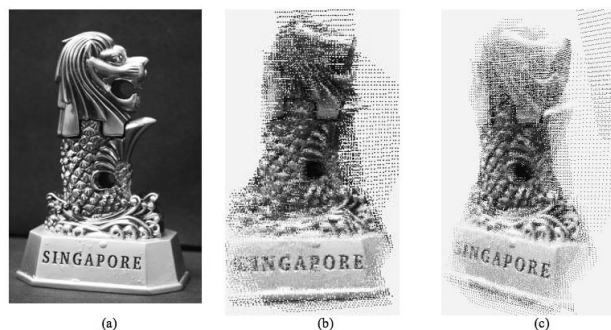
(b)

Gbr. 17 Rekonstruksi objek bangunan Herz-Jesu: (a) Hasil rekonstruksi yang dilakukan oleh Christoph Strecha [19]; (b) Hasil rekonstruksi dengan menggunakan teknik komputasi fotografi pada penelitian ini.

B. Hasil Rekonstruksi Miniatur Patung Merlion

Hasil percobaan berikutnya adalah rekonstruksi objek miniatur Merlion Singapore. Gbr. 18 menunjukkan perbandingan antara objek riil dan objek Merlion hasil rekonstruksi. Gbr. 18.a adalah objek riil. Gbr. 18.b adalah model 3-D hasil rekonstruksi dilihat dari depan, dan Gbr. 18.c menunjukkan model 3-D yang dilihat dari 45° searah jarum jam.

Dari titik-titik fitur yang terdeteksi pada citra Merlion, dihasilkan titik-titik 3-D sebanyak 519 titik 3-D yang sudah terekonstruksi dalam bidang-bidang primitif. Dengan demikian, hanya sekitar 10% titik-titik fitur dari masing-masing citra yang digunakan untuk proses rekonstruksi 3-D objek Merlion. Dalam percobaan ini, kami menggunakan algoritma RANSAC dengan jumlah iterasi sebesar 300 kali. Secara visual, teknik komputasi fotografi kami mampu melakukan rekonstruksi objek Merlion. Pada bagian alas Merlion (tulisan “Singapore”), rekonstruksi 3-D dengan teknik komputasi fotografi bisa dilakukan dengan baik. Namun demikian, masih ada beberapa bagian yang tertutup oleh *noise*, terutama di bagian kepala Merlion. Kami menganalisis, kerumitan tekstur Merlion menjadi salah satu sebab munculnya *noise*. Perlu dilakukan penelitian lebih lanjut, terkait dengan posisi pengambilan citra Merlion dan jumlah iterasi algoritma RANSAC yang diperlukan untuk menghasilkan model 3-D objek Merlion yang lebih baik.



Gbr. 18 Rekonstruksi objek miniatur Merlion Singapore: (a) Objek asli; (b) Hasil rekonstruksi 3-D menggunakan teknik komputasi fotografi dengan iterasi RANSAC 300 kali; (c) Hasil rekonstruksi 3-D setelah model 3-D diputar 45° searah jarum jam.

IV. KESIMPULAN DAN SARAN

Pada penelitian kali ini, rekonstruksi objek 3-D secara visual dengan perangkat keras berbiaya rendah dengan teknik komputasi fotografi berbasis algoritma DLT telah dilaksanakan. Percobaan dilakukan dengan memanfaatkan objek miniatur Merlion Singapore dan bangunan Herz-Jesu. Dari hasil percobaan, sekitar 10-20% dari seluruh titik-titik fitur 2-D terdeteksi dimanfaatkan untuk melakukan rekonstruksi titik 3-D. Dengan demikian, penggunaan citra 2-D dengan resolusi besar akan memperbanyak terdeteksinya titik-titik fitur 2-D dan dengan sendirinya akan meningkatkan jumlah titik-titik 3-D yang dimanfaatkan untuk melakukan proses rekonstruksi objek 3-D. Selain itu, jumlah iterasi algoritma RANSAC sangat berpengaruh terhadap hasil akhir model 3-D. Dari hasil percobaan dapat ditarik kesimpulan bahwa teknik komputasi fotografi dapat digunakan untuk melakukan rekonstruksi objek 3-D secara visual, baik pada objek berukuran kecil (miniatur) maupun objek yang berupa bangunan.

Penelitian lebih lanjut dapat difokuskan pada optimalisasi jumlah titik-titik 2-D pada citra yang digunakan untuk proses rekonstruksi sehingga mencapai sebagian besar (70-90%) dari jumlah keseluruhan titik yang terdeteksi algoritma SIFT. Selain itu, pemodelan dan proses teksturing dapat disesuaikan dengan tingkat kerumitan permukaan objek 3-D yang

dideteksi untuk menghasilkan model 3-D dengan tingkat kedetailan lebih tinggi. Dengan mengembangkan algoritma yang tepat untuk perangkat berbiaya rendah, diharapkan hasil penelitian ini dapat memberikan manfaat dan dipraktikkan secara langsung, terutama untuk membantu menjelaskan konsep sintesa model 3-D dari citra 2-D pada mata kuliah pengolahan citra, serta sebagai dasar pengembangan sistem 3-D scanner berbiaya rendah untuk objek-objek rigid berukuran kecil maupun besar.

REFERENSI

- [1] P. Young and G. Wadge, "FLOWFRONT: Simulation of a lava flow," *Computers & Geosciences*, vol. 16, pp. 1171-1191, 1990.
- [2] M. J. O'Sullivan, K. Pruess, and M. J. Lippmann, "State of the art of geothermal reservoir simulation," *Geothermics*, vol. 30, pp. 395-429, 2001.
- [3] N. Ezquerra, L. de Braal, E. Garcia, C. Cooke, and E. Krawczynska, "Interactive, knowledge-guided visualization of 3D medical imagery," *Future Generation Computer Systems*, vol. 15, pp. 59-73, 1999.
- [4] S. E. Mahmoudi, A. Akhondi-Asl, R. Rahmani, S. Faghieh-Roohi, V. Taimouri, A. Sabouri, and H. Soltanian-Zadeh, "Web-based interactive 2D/3D medical image processing and visualization software," *Computer Methods and Programs in Biomedicine*, vol. 98, pp. 172-182, 2010.
- [5] E. Martínez and C. Torras, "Qualitative vision for the guidance of legged robots in unstructured environments," *Pattern Recognition*, vol. 34, pp. 1585-1599, 2001.
- [6] M. Y. Kim and H. Cho, "An active trinocular vision system of sensing indoor navigation environment for mobile robots," *Sensors and Actuators A: Physical*, vol. 125, pp. 192-209, 2006.
- [7] Joo and M. Isabel, "Mobile robot localisation on reconstructed 3D models," *Robotics and Autonomous Systems*, vol. 31, pp. 17-30, 2000.
- [8] W. S. Lee and N. Magnenat-Thalmann, "Fast head modeling for animation," *Image and Vision Computing*, vol. 18, pp. 355-364, 2000.
- [9] S. Horna, D. Meneveaux, G. Damiand, and Y. Bertrand, "Consistency constraints and 3D building reconstruction," *Computer-Aided Design*, vol. 41, pp. 13-27, 2009.
- [10] A. Koutsoudis, F. Arnaoutoglou, and C. Chamzas, "On 3D reconstruction of the old city of Xanthi. A minimum budget approach to virtual touring based on photogrammetry," *Journal of Cultural Heritage*, vol. 8, pp. 26-31.
- [11] F. Bruno, S. Bruno, G. De Sensi, M.-L. Luchi, S. Mancuso, and M. Muzzupappa, "From 3D reconstruction to virtual reality: A complete methodology for digital archaeological exhibition," *Journal of Cultural Heritage*, vol. 11, pp. 42-49, 2010.
- [12] M. Sonka, V. Hlavac, and R. Boyle, "3D Vision," in *Image Processing, Analysis, and Machine Vision*, H. Gowans, Ed., 3rd ed. Toronto: Thomson, pp. 592-594, 2008.
- [13] H. Anton & R.C. Busby, "Singular Value Decomposition" in *Contemporary Linear Algebra*, 1st ed. New Jersey: John Wiley & Sons, Inc. pp. 502-516, 2003.
- [14] H. Anton & R.C. Busby, "Q-R Decomposition: Householder Transformation" in *Contemporary Linear Algebra*, 1st ed. New Jersey: John Wiley & Sons, Inc. pp. 417-426, 2003.
- [15] D.G. Lowe, "Object recognition from local scale-invariant features", *Proceedings of the International Conference on Computer Vision*, pp. 1150-1157, 1999.
- [16] C. Strecha. (2008, 5 February). *Herz-Jesu P-8 Datasets*. Available: http://cvlab.epfl.ch/~strecha/multiview/herzjesu_dense_large.html.
- [17] C. Strecha, W. von Hansen, L. Van Gool, P. Fua, U. Thoennessen, "On Benchmarking Camera Calibration and Multi-View Stereo for High Resolution Imagery", *CVPR*, 2008.
- [18] V. Vezhnevets and A. Velizhev. (2008, 10 February). *GML C++ Camera Calibration Toolbox*. Available: <http://research.graphicon.ru/calibration/gml-c++-camera-calibration-toolbox.html>.
- [19] OpenSceneGraph. (2011, 10 February). *OpenSceneGraph Official Homepage*. Available: <http://www.openscenegraph.org/projects/osg>
- [20] Y.I. Abdel-Aziz & H.M. Karara, "Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry", in *Proceeding of the Symposium on Close Range Photogrammetry*, pp. 1-18, 1971.
- [21] S. Wibirama, S. Tungjitkusolmun, and C. Pintavirooj, "Dual-Camera Acquisition for Accurate Measurement of Three-Dimensional Eye Movements," in *IEEJ Transactions on Electrical and Electronic Engineering*, Vol.8, No. 3, 2012.
- [22] S. Wibirama, S. Tungjitkusolmun, C. Pintavirooj, and K. Hamamoto, "Dual Cameras Acquisition For Three-Dimensional Eye-Motion Tracking" in *Proceeding of 2nd Biomedical Engineering International Conference (BMEiCON 2009)*, Thailand, pp.18-19, 2009.
- [23] G. Bradski and A. Kaehler, "Camera Models and Calibration," in *Learning OpenCV: Computer Vision with The OpenCV Library*, 1st ed California: O'Reilly, pp. 370-375, 2008.
- [24] M. Sonka, V. Hlavac, and R. Boyle, "Convex Hull," in *Image Processing, Analysis, and Machine Vision*, H. Gowans, Ed., 3rd ed. Toronto: Thomson, pp. 360-364, 2008.
- [25] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, pp. 381-395, 1981.