

PEMANFAATAN FILTER DALAM PENINGKATAN HASIL SKELETAL TRACKING PADA KINECT

Erick Pranata ⁽¹⁾, Lukman Zaman P. C. S. W. ⁽²⁾

Sekolah Tinggi Teknik Surabaya

e-mail: erick@stts.edu⁽¹⁾, luqmanz@stts.edu⁽²⁾

ABSTRAK

Skeletal tracking merupakan teknik yang digunakan untuk menentukan posisi sendi manusia berdasarkan citra yang diperoleh dari sensor. Teknik ini sudah mulai diterapkan pada kamera RGB, termasuk sensor terbaru milik Microsoft yang disebut sebagai Kinect. Dengan menggunakan Microsoft Kinect SDK, posisi sendi seseorang yang dideteksi oleh sensor dapat dideteksi secara otomatis. Namun, hasil skeletal tracking tersebut masih belum baik. Penelitian ini mengusulkan teknik untuk meningkatkan hasil deteksi sendi dengan menggunakan dua buah Kinect. Peningkatan hasil deteksi dilakukan dengan filtering pada Kinect Utama menggunakan Moving Average dan Kalman Filter yang dimodifikasi, dilengkapi dengan dukungan data dari Kinect Pendukung yang diletakkan pada berbagai konfigurasi untuk mendeteksi pose-pose yang sulit ditangkap oleh satu Kinect saja. Penelitian ini membuktikan bahwa filtering dengan Moving Average dan Kalman Filter yang dimodifikasi mampu meningkatkan kualitas deteksi sendi secara visual. Hasil deteksi pun menjadi lebih baik dengan penggunaan Kinect Pendukung sebagai data bantuan, dimana sendi dengan status tracked mampu mencapai 82.93% dari seluruh posisi sendi yang ada.

Kata kunci: Kinect, Joint, Kalman Filter, OGRE, NUI, Mocap, Skeletal Tracking

ABSTRACT

Skeletal tracking is technique that can be used to determine human joint positions according to images gathered by a sensor. This technique has been used on RGB camera, including the newest sensor from Microsoft called Kinect. By utilizing Microsoft Kinect SDK, human joint positions can be detected by the sensor automatically. However, the skeletal tracking result which is produced by Microsoft Kinect SDK is still not good enough. This study proposes techniques to increase the result of joint detection by using two Kinects. Detection result improvement is done by implementing Moving Average filter and Modified Kalman Filter to data recorded, accompanied by second Kinect which is configured in many ways to detect poses that cannot be detected by a single Kinect alone. This study proves that filtering using Moving Average and Modified Kalman Filter is able to improve joint detection visually. Detection yields a better result by using second Kinect as helper, where joint positions produced are tracked up to 82.93% of all existing joints.

Keywords: Kinect, Joint, Kalman Filter, OGRE, NUI, Mocap, Skeletal Tracking.

PENDAHULUAN

Motion Capture (akuisisi gerakan) yang biasa dikenal dengan singkatan mocap, merupakan proses perekaman gerakan secara langsung yang kemudian diterjemahkan dari *live performance* (pergelaran langsung) menjadi *digital performance* (pergelaran digital). Mocap digunakan diberbagai bidang, terutama animasi.

Teknologi mocap yang telah diperkenalkan sejak abad ke-19 ini dapat diimplementasikan dengan 2

teknik, yaitu Marker-based Optical Capture dan Markerless Motion Capture. Untuk teknik Markerless, dibutuhkan teknik Computer Vision dengan dukungan hardware yang memadai, seperti Kinect.

Kinect memiliki kemampuan untuk mendeteksi *skeleton* (struktur tulang manusia) yang direkamnya dengan error deteksi *joint* (sendi) sekitar 10 cm; juga error saat terdapat gerakan dimana terdapat posisi tubuh yang menutupi bagian tubuh yang lain dan error yang disebabkan adanya objek lain dalam

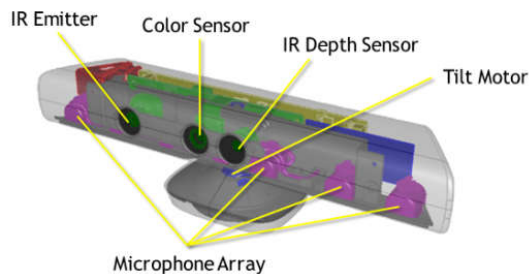
scene (tempat perekaman) (Han et al., 2013). Error tersebut merupakan error yang disebabkan oleh adanya *noise* (gangguan) yang tidak memiliki model matematis, sehingga satu-satunya cara untuk mengatasinya adalah dengan menggunakan filter.

Dalam penelitian ini juga diusulkan penggunaan dua buah Kinect dalam satu *scene* untuk meningkatkan akurasi data yang diperoleh. Dengan pendekatan ini, terbentuk *noise* terdefinisi, yaitu *noise* yang disebabkan oleh penggabungan dua buah Kinect. Dengan demikian, barulah *noise* tersebut dapat diatasi dengan transformasi 3 dimensi, agar data yang dihasilkan dapat mendukung mocap *markerless*.

KINECT

Kinect merupakan salah satu hardware yang mendukung Natural User Interface (NUI) yang sudah mulai digagas sejak tahun 70-an (Jarrett Webb., 2012). Kinect merupakan formulasi dari kompetisi *gaming console* antara Microsoft dengan pesaing-pesaingnya. Namun dalam perkembangannya, Kinect banyak membuka kesempatan bagi para developer untuk mengimplementasikannya tidak hanya dalam dunia *gaming*, salah satunya pada *motion capture*.

Kinect merupakan hardware yang mampu menerima input audio visual. Kinect memiliki microphone array untuk menerima input audio dan ColorSensor yang dilengkapi dengan IR Depth Sensor untuk menangkap elemen visual. Susunan hardware Kinect dapat dilihat pada Gambar 1.



Gambar 1. Komponen Penyusun Kinect (Microsoft., 2015)

Setiap komponen penyusun Kinect tersebut membuat Kinect mampu mendeteksi skeleton dari manusia yang direkamnya. Selain komponen yang telah ditunjukkan pada Gambar 1, Kinect juga dilengkapi dengan sebuah motor yang terletak diantara badan dan alas Kinect. Motor ini digunakan

untuk mengubah sudut pandang Kinect secara vertikal.

Hingga 2015, Microsoft telah meluncurkan 2 versi Kinect, yaitu Kinect V1 dan Kinect V2. Kinect V1 dapat mendeteksi hingga 6 orang, namun hanya 2 orang terdekat saja yang dapat diekstrak skeletonnya. Sedangkan Kinect V2 sudah mampu mendeteksi 6 skeleton secara detail. Dapat disimpulkan Kinect V2 memiliki kemampuan lebih baik dari Kinect V1. Hal ini terlihat dari kapabilitas Color Sensor dan IR Depth Sensor yang lebih baik. Pada penelitian ini Kinect yang digunakan adalah Kinect V2.

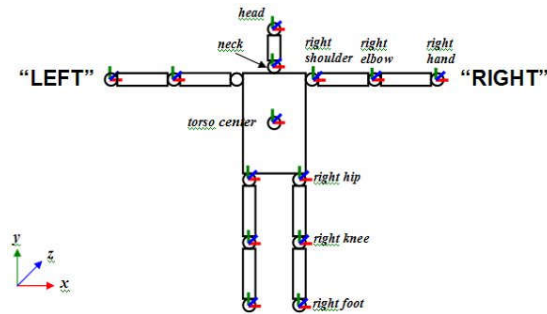
Feature	Kinect for Windows 1	Kinect for Windows 2
Color Camera	640 x 480 @30 fps	1920 x 1080 @30 fps
Depth Camera	320 x 240	512 x 424
Max Depth Distance	~4.5 M	~4.5 M
Min Depth Distance	40 cm in near mode	50 cm
Horizontal Field of View	57 degrees	70 degrees
Vertical Field of View	43 degrees	60 degrees
Tilt Motor	yes	no
Skeleton Joints Defined	20 joints	26 joints
Full Skeletons Tracked	2	6
USB Standard	2.0	3.0
Supported OS	Win 7, Win 8	Win 8
Price	\$299	TBD

Gambar 2. Perbandingan Kinect V1 dan Kinect V2

Berbagai pengembangan aplikasi yang memanfaatkan Kinect pada beberapa sistem operasi memerlukan *library* yang dapat menghubungkan Kinect dengan sistem operasi tersebut pada suatu bahasa pemrograman. Terdapat banyak *library* yang dapat digunakan, namun *library* yang paling umum digunakan adalah Microsoft Kinect SDK dan OpenNI/NITE.

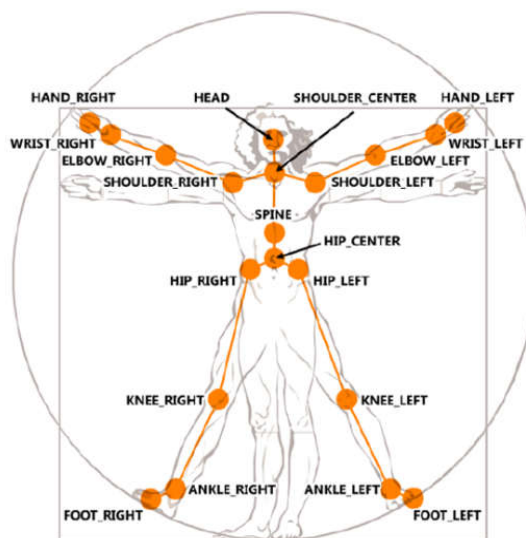
OpenNI/NITE merupakan SDK untuk *3D Sensing Hardware* (hardware yang digunakan untuk mengubah live performance menjadi digital performance) yang dikembangkan di 2 platform utama, yaitu Windows dan Linux. OpenNI merupakan framework open source yang berada dalam lingkungan C++. NITE merupakan middleware yang dikembangkan dengan OpenNI, yang dapat digunakan untuk mengembangkan aplikasi berbasis NUI. Middleware ini mendukung deteksi skeleton, baik deteksi tangan, maupun deteksi seluruh tubuh. Algoritma yang diimplementasikan memanfaatkan IR Depth Color,

Color Sensor dan Audio Sensor. Jumlah joint yang dapat dideteksi oleh NITE berbeda dengan Microsoft Kinect SDK. Terdapat 15 joint yang dideteksi NITE, sementara Microsoft Kinect SDK mendeteksi 20 joint. Belum ada definisi formal mengenai jumlah joint ideal yang harus dideteksi oleh 3D Sensing Hardware.



Gambar 3. Hasil Skeletal Tracking dengan NITE

Microsoft Kinect SDK merupakan library yang dikembangkan oleh Microsoft khusus untuk sistem operasi Windows. Berbeda dengan OpenNI/NITE, Microsoft Kinect SDK tidak bersifat open source. Selain menyediakan Kinect SDK, Microsoft juga menyediakan beberapa contoh aplikasi yang siap dikembangkan, berkaitan dengan fitur-fitur yang dimiliki oleh Kinect. Microsoft menyebut aplikasi ini dengan istilah Kinect for Windows Developer Toolkit. Dalam aplikasi tersebut juga disediakan aplikasi yang dapat digunakan untuk merekam hasil stream Kinect, yaitu Kinect Studio. Berdasarkan jumlah joint yang dihasilkan, hasil dari Microsoft Kinect SDK ini menyajikan informasi joint yang lebih akurat dibandingkan OpenNI.



Gambar 4. Skeletal Tracking Microsoft Kinect SDK

PERUMUSAN MASALAH

Terdapat dua masalah utama yang melatarbelakangi penelitian ini, yaitu adanya noise pada hasil skeletal tracking dan adanya tujuan untuk meningkatkan akurasi dari hasil skeletal tracking dengan memanfaatkan 2 buah kinect.

1. NOISE PADA HASIL SKELETAL TRACKING

Skeletal tracking pada Kinect SDK menggunakan algoritma yang dikembangkan oleh Shotton et al. pada tahun 2010. Algoritma ini berbasis *machine learning* dengan pendekatan *object recognition*. Untuk pembentukan algoritma ini, pertama-tama Shotton mengumpulkan dataset dalam bentuk gambar yang dilengkapi dengan informasi Depth Image yang diperoleh dari IR Depth Sensor. Dari 500.000 lebih gambar yang dikumpulkan dari serangkaian kegiatan manusia, dilakukan *preprocessing* berupa penghapusan serangkaian gambar yang memiliki kemiripan. Hal ini bertujuan supaya sistem tidak perlu memproses sejumlah gambar yang mirip. Proses ini dilakukan dengan memanfaatkan algoritma *furthest neighbor clustering* (T. Gonzalez. 1985).

Dataset dalam bentuk Depth Image ini kemudian digiring menjadi *body parts representation data*. Data ini dibentuk dengan memanfaatkan fitur dari Depth Image dengan mengklasifikasikan setiap pixel ke bagian tubuh tertentu dengan *randomized decision tree* (Lepetit et al., 2005). Setiap pixel yang telah digolongkan ini akan menciptakan area-area yang menyatakan bagian tubuh. Dan untuk menentukan posisi sendi, perlu diketahui terlebih dahulu pusat setiap area tersebut. Di sinilah *Mean Shift* (Dorin Comaniciu et al., 2002) berperan.

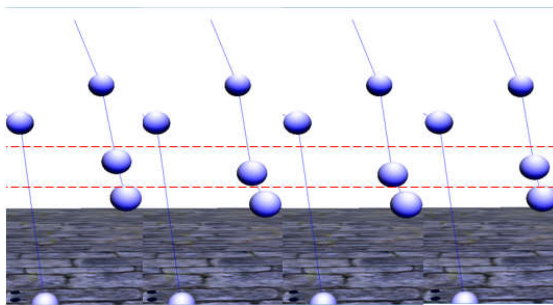


Gambar 5. Sendi yang Dihasilkan dari Intermediate Body Parts Representation

Informasi pusat body part akan membantu dalam menentukan posisi sendi. Sebagai contoh, dengan informasi badan kiri atas, badan kanan atas, badan kiri bawah, dan badan kanan bawah, dapat diidentifikasi sebuah spine (sendi tulang belakang), seperti terlihat pada Gambar 5.

Proses penentuan sendi ini berjalan dengan cepat (sekitar 200 fps). Hal ini membuat algoritma ini dapat digunakan pada level interaktif sebagai kontroler permainan. Namun seiring meningkatnya jenis aplikasi Kinect, algoritma ini dituntut untuk lebih akurat.

Kinect SDK didesain untuk melakukan skeletal tracking untuk posisi menghadap ke depan, akan tetapi dalam sejumlah frame yang runut, terdapat posisi joint yang seharusnya diam, tampak terus bergerak. Hal inilah yang akan berusaha diatasi dengan menggunakan filter. Memanfaatkan Kalman Filter, diupayakan sinyal-sinyal yang ditemukan dapat dibuat konvergen sehingga *jitter* yang tampak dapat dikurangi.



Gambar 6. Jitter pada Skeletal Tracking

2. JOINT TERSEMBUNYI

Gerakan-gerakan yang dilakukan pada sebuah live performance memiliki kompleksitas yang cukup tinggi. Terkadang, beberapa joint tertutup oleh bagian tubuh, sehingga Kinect SDK hanya mampu memberikan status INFFERED pada joint-joint seperti itu.

Kinect SDK didesain agar dapat mengenali skeleton bila dipandang dari depan. Dengan demikian, status INFFERED yang diberikan oleh Kinect SDK perlu disempurnakan dengan dukungan data lainnya. Berdasarkan temuan inilah, diperlukan suatu ide untuk menyempurnakan data tersebut.

Penggunaan Kinect kedua merupakan salah satu ide untuk melengkapi data yang sudah ada. Dalam sebuah perekaman live performance, digunakan dua buah Kinect secara bersamaan. Data dari kedua

Kinect ini nantinya akan digabungkan untuk menghasilkan data posisi joint yang lebih akurat jika dibandingkan dengan 1 kinect saja.

STRATEGI PENELITIAN

Hipotesis yang berusaha dijawab oleh penelitian ini adalah hasil smoothing yang secara visual lebih baik dan joint yang 90% terdeteksi dengan penggabungan data dari dua Kinect. Dengan demikian, secara umum, penelitian yang dikerjakan pada penelitian ini terbagi atas 3 tahap utama, yaitu Tahap Perekaman, Tahap Smoothing, dan Tahap Penggabungan.

Tahap Perekaman merupakan tahap pengumpulan data dari kedua Kinect. Pada tahap ini, diuji beberapa konfigurasi posisi Kinect terhadap model yang akan direkam. Kedua Kinect dijalankan secara bersamaan untuk mendapatkan data rekaman yang selaras.

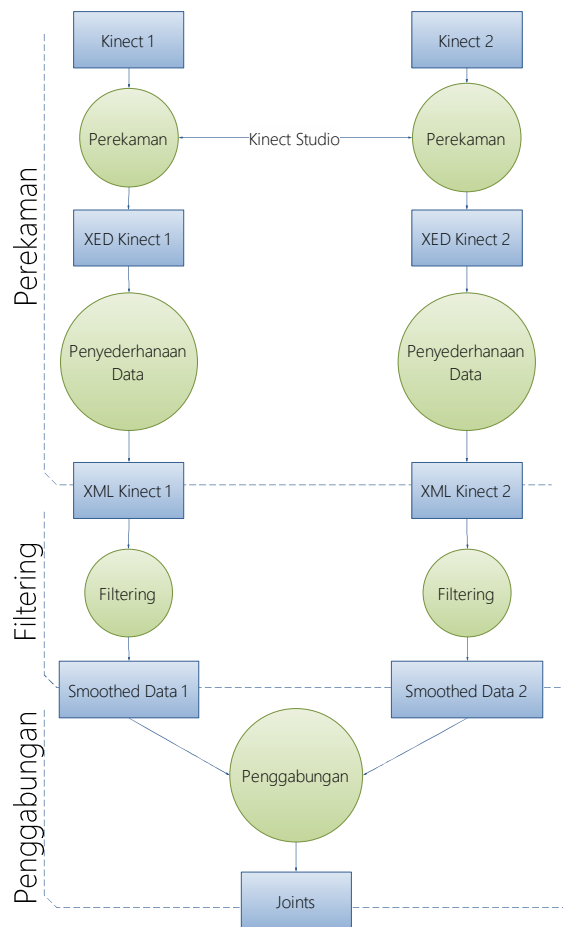
Tahap Smoothing merupakan tahap yang dilakukan untuk memperhalus data yang diperoleh. Tanpa tahap ini, joints yang diperoleh dari skeletal tracking akan tampak *jittery*. Sejak versi 1.5, Kinect SDK telah menyediakan filter *built-in* dengan metode Holt Double Exponential Smoothing yang umumnya digunakan untuk analisis statistik data ekonomi (Clemente Giorio, 2015). Penelitian ini hanya mencoba menerapkan Kalman Filter (Bilgin Esme, 2015) dan filter Moving Average.

Tahap Penggabungan merupakan tahap yang dilakukan untuk menggabungkan kedua data Kinect yang telah dihaluskan. Penggabungan menggunakan model matematika yang sebagian besar tersusun atas tranformasi 3 dimensi. Penggabungan ini menjadikan konfigurasi posisi Kinect sebagai parameter input dalam model, dan menghasilkan keluaran berupa kumpulan joint gabungan.

Data kedua Kinect pada mulanya direkam dengan menggunakan Kinect Studio. Data yang direkam ini akan menghasilkan sepasang dokumen dengan ekstensi .xed, yang dapat dijalankan kembali tanpa harus melakukan live performance. Dengan diubahnya data Kinect menjadi .xed, proses pengolahan data pada penelitian ini menjadi lebih sederhana.

Sesudah data diubah ke dalam bentuk .xed, tahap berikutnya adalah pengubahan data menjadi bentuk yang lebih portabel, yaitu file XML. Dengan

diubahnya data Kinect menjadi XML, Kinect sudah tidak diperlukan lagi. Berkenaan dengan perubahan ini, data utama yang disimpan adalah posisi dan status untuk tiap joint.



Gambar 7. Arsitektur Penelitian

Data yang diperoleh selanjutnya diperhalus dengan menggunakan filter Moving Average atau Kalman Filter yang dimodifikasi. Penentuan filter yang digunakan bersifat kualitatif, karena filter ini hanya bertujuan untuk menghapus jitter pada rangkaian posisi joint yang diperoleh.

Joint yang sudah diproses inilah yang kemudian digabungkan menjadi 1 himpunan data. Penggabungan bersifat offline, sehingga gabungan data ini pun disimpan dalam sebuah file yang nantinya dapat digunakan untuk berbagai keperluan. Penelitian ini hanya berhenti sampai data gabungan offline, namun tidak menutup kemungkinan teknik yang digunakan pada penelitian ini dapat digunakan untuk memproses data yang diperoleh secara *real*.

PEREKAMAN DAN FILTERING

Perekaman live performance dengan dua Kinect memerlukan beberapa kali perekaman dengan beberapa konfigurasi. Live performance yang dilakukan oleh model untuk sebuah perekaman didefinisikan dalam table 1, dengan referensi gerakan dasar Senam Kesegaran Jasmani (SKJ). Sementara konfigurasi setting kedua Kinect didefinisikan pada table 2. Terdapat 4 variasi konfigurasi dan 2 jenis live performance. Sehingga, jumlah data yang direkam untuk sebuah Kinect adalah 8 rekaman. Dengan demikian, total data rekaman untuk kedua Kinect adalah 16 rekaman. Perekaman dilakukan pada 17 Januari 2015 di dalam ruang tertutup dengan luas yang cukup lebar.

Tabel 1. Gerakan Dasar SKJ

Tubuh Atas	Tubuh Bawah
Merebahkan kepala ke kiri	Mengangkat kaki kanan
Merebahkan kepala ke kanan	Mengangkat kaki kiri
Menundukkan kepala	Melangkahkan kaki kanan ke depan
Mengangkat bahu	Melangkahkan kaki kiri ke depan
Mengangkat lengan kanan ke depan	Melangkahkan kaki kanan ke belakang
Mengangkat lengan kiri ke depan	Melangkahkan kaki kiri ke belakang
Mengangkat lengan kanan ke atas	Mengangkat kaki kanan ke samping
Mengangkat lengan kiri ke atas	Mengangkat kaki kiri ke samping
Mengangkat lengan kanan ke samping	Menggerakkan pinggul ke kanan
Mengangkat lengan kiri ke samping	Menggerakkan pinggul ke kiri
Tepuk Tangan	Berputar 90° ke kanan
Membungkuk	Berputar 90° ke kiri
Memutar badan atas ke kanan	
Memutar badan atas ke kiri	

Hasil rekaman yang diperoleh masih bersifat raw (mentah). Hal ini disebabkan karena bagian awal dan bagian akhir rekaman tidak masuk dalam live performance. Oleh karena itu, rekaman tersebut perlu diproses terlebih dahulu, agar lebih sesuai dengan rekaman yang diharapkan.

Tabel 2. Konfigurasi Setting

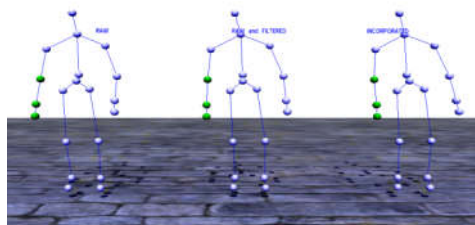
Depan dan 90° di Kanan	Depan dan 45° di Kanan
Posisi Kinect–Posisi Model=3 m	Posisi Kinect–Posisi Model=3 m
Posisi Kinect–Posisi Lantai=1 m	Posisi Kinect–Posisi Lantai=1 m
Tilt motor = 0°	Tilt motor = 0°
Posisi Kinect–Posisi Model=3 m	Posisi Kinect–Posisi Model=3 m
Posisi Kinect–Posisi Lantai=1,5m	Posisi Kinect–Posisi Lantai=1,5m
Tilt motor = -3°	Tilt motor = -3°

Tahap perekaman merupakan tahap dimana mengubah kumpulan data XED yang diperoleh dari hasil perekaman menjadi data XML yang lebih

mudah diolah. Sebelum proses pengubahan data dari XED menjadi XML terlebih dahulu dilakukan proses pemotongan bagian awal dan akhir, yang merupakan bagian residu yang berisi cue yang menunjukkan akhir dari sebuah gerakan serial.

Rekaman yang sudah diproses kemudian satu per satu diubah ke dalam XML. Untuk mengubah rekaman ini, perlu didefinisikan terlebih dahulu struktur data yang dihasilkan oleh Kinect SDK untuk selanjutnya disesuaikan dengan struktur XML yang akan dibentuk. Untuk mempermudah pengerjaan, perlu dibentuk terlebih dahulu *intermediary object* (objek penengah) untuk menyimpan hasil skeletal tracking dari rekaman. Intermediary object yang dihasilkan akan disimpan dalam sebuah *array* (kumpulan data) berdasarkan frame dalam rekaman. Rata-rata jumlah frame untuk live performance Tubuh Atas adalah 2152 frame dan Tubuh Bawah adalah 1847 frame. Array yang berisi object ini lalu disimpan ke dalam file XML. Setiap frame pada file XML berisi 20 joint yang dideteksi oleh Kinect.

Untuk keperluan visualisasi, file XML yang dihasilkan harus dibaca oleh program dan ditampilkan dalam layar 3 dimensi. Selain itu, perlu juga interaksi dengan *input device* agar proses navigasi dapat dilakukan saat visualisasi. OGRE merupakan engine 3 dimensi yang digunakan dalam penelitian ini. Engine ini dapat digunakan dalam bahasa pemrograman C++ dan bersifat open source. Namun, beberapa developer telah menyediakan *wrapper* (perantara) bagi OGRE sehingga dapat digunakan dengan bahasa pemrograman C#.



Gambar 8. Visualisasi Skeleton dengan OGRE

Visualisasi telah berjalan baik, namun secara kualitatif, joint-joint yang ditampilkan bersifat jittery, sehingga mengurangi kualitas skeletal tracking. Bilamana jitter ini tidak ditangani, output yang dihasilkan oleh Kinect ini akan sangat mengurangi kualitas animasi yang dihasilkan pada mocap. Salah satu solusi untuk mengurangi jitter yang muncul adalah dengan menggunakan filtering

(baik dengan filter built-in pada Kinect SDK, Moving Average, maupun Kalman Filter).

1. FILTER BUILT-IN KINECT SDK

Filtering built-in yang disediakan oleh Kinect SDK menggunakan Holt Double Exponential Smoothing. Terdapat 5 parameter yang dapat diatur untuk filter ini (Clemente Giorio, 2015), yaitu:

- *Smoothing*: Meningkatkan nilai parameter ini akan meningkatkan tingkat kehalusan posisi joint yang dihasilkan namun diiringi dengan latency yang tinggi pula.
- *Correction*: Parameter ini menyatakan seberapa cepat pemberian smoothing terhadap hasil skeletal tracking.
- *Prediction*: Menyatakan jumlah frame posisi prediksi untuk joint yang dihasilkan.
- *JitterRadius*: Berguna untuk membatasi pergeseran jitter.
- *MaxDeviationRadius*: Berguna untuk membatasi penyimpangan data dari data raw yang dihasilkan.

2. KALMAN FILTER

Kalman Filter merupakan sebuah model matematika yang menerima sebuah input nilai dan menghasilkan sebuah output estimasi nilai, dimana estimasi ini merupakan nilai yang menuju konvergen. Dengan demikian dapat dikatakan bahwa Kalman Filter dapat digunakan untuk, secara perlahan-lahan, menghasilkan nilai yang konvergen untuk serangkaian data yang terkadang menyimpang. Model matematis ini dinyatakan sebagai berikut:

$$X_k = K_k \cdot Z_k + (1 - K_k) \cdot X_{k-1}$$

dimana:

- X_k : merupakan nilai estimasi
- K_k : menyatakan nilai gain pada frame tersebut
- Z_k : merupakan nilai input
- X_{k-1} : menyatakan nilai estimasi pada frame sebelumnya

Kalman Filter merupakan model matematis yang dijalankan secara iteratif, sebanyak jumlah data yang akan diproses. Setiap iterasi menjalankan proses Time Update dan Measurement Update. Time Update dilakukan untuk menghasilkan nilai estimasi, sementara Measurement Update dilakukan untuk menentukan nilai gain. Formula untuk Time Update dinyatakan sebagai berikut:

$$X_k = X_{k-1}$$

$$P_k = P_{k-1}$$

Sedangkan Measurment Update dinyatakan sebagai berikut:

$$K_k = \frac{P_k}{P_k + R}$$

$$X_k = X_k + K_k(Z_k - X_k)$$

$$P_k = (1 - K_k)P_k$$

Nilai R , diperoleh dengan menghitung standar deviasi per 30 frame. Penentuan nilai 30 berdasar pada perilaku Kinect yang pada umumnya menghasilkan pembacaan dengan kecepatan 30 frame per detik. Nilai R dihitung dengan menggunakan n bernilai 30.

$$R = \sqrt{\frac{\sum_{i=1}^n (\bar{x} - x_i)^2}{n - 1}}$$

Modifikasi yang dilakukan adalah dengan mengubah nilai covariance P_k yang diumpangkan saat menentukan nilai gain menjadi nilai konstan, yaitu 0,5. Perubahan ini menyebabkan nilai gain relatif konstan, walaupun terjadi sedikit perubahan per 30 frame yang disebabkan oleh perubahan standar deviasi R .

Modifikasi ini menghasilkan tampilan visual yang lebih baik. Konvergensi terjadi secara lokal, dan tidak terdapat spike setiap 30 frame. Penelitian tahap filtering dihentikan sampai tahap ini, karena hasil visual yang diperoleh sudah baik dan menjawab permasalahan yang muncul akibat data Kinect yang jittery.

PENGGABUNGAN 2 BUAH KINNECT

Penggabungan kedua data Kinect dilakukan dengan mentransformasikan data Kinect Pendukung agar selaras dengan Kinect Utama. Data tersebut digabungkan dengan menggunakan transformasi 3 dimensi. Penggabungan memperhatikan konfigurasi pemasangan Kinect, seperti dijabarkan pada Tabel 3.

Faktor-faktor yang perlu diperhatikan dalam penggabungan adalah:

- Jarak Kinect Utama dan Kinect Pendukung terhadap model utama yang direkam. Informasi ini sangat penting untuk menentukan translasi data yang dihasilkan oleh Kinect Pendukung.
- Sudut yang terbentuk antara Kinect Utama dan Kinect Pendukung. Data yang dihasilkan oleh Kinect Pendukung harus dirotasikan agar selaras dengan data Kinect Utama.

Tabel 3. Daftar Konfigurasi Perekaman

Nama	Deskripsi Konfigurasi
Konfigurasi 1	Kinect Utama di depan model Kinect Pendukung di kanan model Sudut antara Kinect Utama dan Kinect Pendukung adalah 90°. Tinggi masing-masing Kinect 1 meter.
Konfigurasi 2	Kinect Utama di depan model Kinect Pendukung di kanan model Sudut antara Kinect Utama dan Kinect Pendukung adalah 90°. Tinggi masing-masing Kinect 1,4 meter.
Konfigurasi 3	Kinect Utama di depan model Kinect Pendukung di kanan model Sudut antara Kinect Utama dan Kinect Pendukung adalah 45°. Tinggi masing-masing Kinect 1 meter.
Konfigurasi 4	Kinect Utama di depan model Kinect Pendukung di kanan model Sudut antara Kinect Utama dan Kinect Pendukung adalah 45°. Tinggi masing-masing Kinect 1,4 meter.

Kedua faktor tersebut menjadi input dalam model penggabungan, untuk membentuk matriks tranformasi gabungan. Matriks tranformasi gabungan ini akan diterapkan pada seluruh joint pada tiap frame.

$$T = \text{Trans2} \text{ Rot } \text{Trans1} \text{ Posisi}$$

Setelah data Kinect Pendukung ditransformasi agar selaras dengan Kinect Utama, langkah berikutnya adalah mengganti joint-joint dengan status yang belum TRACKED pada Kinect Utama. Penggantian tersebut mengikuti heuristik yang diterapkan berdasarkan sifat dasar Kinect.

Algoritma Penggabungan Posisi Joint

```

1: Jika status_joint_utama ==
2:   NOT_TRACKED maka
3:   joint_terpilih = joint_pendukung
4: Else, jika status_joint_utama ==
5:   INFERRED maka
6:   Jika status_joint_pendukung ==
7:     TRACKED dan distance (utama,
8:     pendukung) <= 0.1 meter maka
9:     Joint_terpilih = joint_pendukung

```

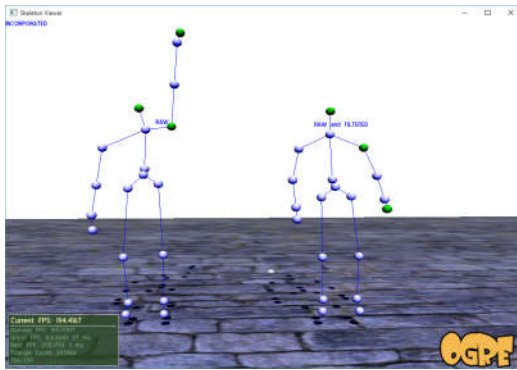
Secara umum, joint yang akan dipilih sebagai output akhir adalah joint yang memiliki status NOT_TRACKED dan INFERRED. Joint yang dipilih mula-mula berasal dari Kinect Pendukung. Namun setelah melalui proses penggabungan, ada kemungkinan nilai joint tersebut digantikan oleh data dari Kinect Pendukung. Joint tersebut akan digantikan dengan data dari Kinect Pendukung apabila status joint pada Kinect Utama adalah NOT_TRACKED. Namun bila status joint dari Kinect Utama adalah INFERRED, maka joint tersebut akan diganti oleh joint dari Kinect Pendukung dengan syarat: status joint Kinect

Pendukung harus TRACKED dan jarak antara joint Kinect Utama dan Kinect Pendukung harus lebih kecil atau sama dengan 0,1 meter.

Perhitungan jarak antara kedua titik menggunakan Euclidian Distance. Penggunaan formula ini didasarkan pada bentuk data posisi joint, yaitu koordinat ruang. Euclidian Distance merupakan formula yang paling cocok untuk kasus tersebut.

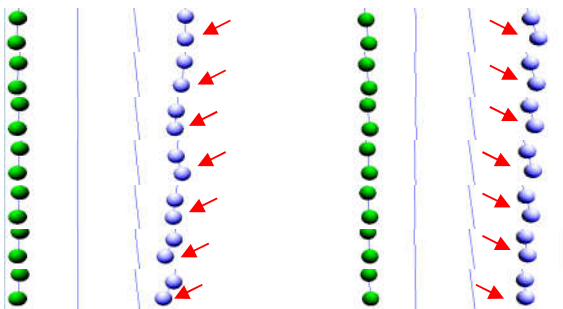
UJICOB

Implementasi Kalman Filter standar menunjukkan delay yang cukup signifikan. Joint-joint yang difilter mengikuti bentuk skeleton rekaman awal sekonvergen mungkin. Dengan bentuk seperti ini, skeletal tracking dengan Kalman Filter standar tidak dapat digunakan untuk memperbaiki hasil skeletal tracking.



Gambar 9. Skeletal Tracking dengan Kalman Filter Standar

Implementasi skeletal tracking dengan Kalman Filter yang dimodifikasi menghasilkan tampilan visual yang lebih baik. Terlihat bahwa pada keadaan standar, posisi joint HAND_RIGHT tersebut bergerak pada posisi yang terlihat kurang wajar, namun dikoreksi dengan baik oleh Kalman Filter yang telah dimodifikasi. Dengan hasil visual seperti ini, Kalman Filter yang dimodifikasi terbukti dapat memperbaiki hasil skeletal tracking Kinect SDK.



Gambar 10. Skeletal Tracking dengan Kalman Filter yang Dimodifikasi

Frame ke-4 menunjukkan perubahan posisi joint yang cukup signifikan. Namun, filter mempertahankan posisi joint tersebut agar tidak menyimpang terlalu jauh dari posisi sebelumnya. Secara visual, hasil ini baik dan secara umum mengurangi jitter pada skeletal tracking.

Tabel 4. Perbandingan Status Joint

Live Performance	TOTAL	TRACKED	INFERRED	NOT TRACKED	TRACKED/TOTAL
Tubuh Atas + Konfigurasi 1	42020	38483	3077	460	38483/42020 (91.58%)
Tubuh Bawah + Konfigurasi 1	37600	34784	2296	520	34784/37600 (92.51%)
Tubuh Atas + Konfigurasi 2	41280	37467	3253	560	37467/41280 (90.76%)
Tubuh Bawah + Konfigurasi 2	36960	34238	2142	580	34238/36960 (92.64%)
Tubuh Atas + Konfigurasi 3	48240	42849	4931	460	42849/48240 (88.82%)
Tubuh Bawah + Konfigurasi 3	34860	32851	1549	460	32851/34860 (94.24%)
Tubuh Atas + Konfigurasi 4	39380	36039	2841	500	36039/39380 (91.52%)
Tubuh Bawah + Konfigurasi 4	37420	35149	1771	500	35149/37420 (93.93%)
Rata-rata Akurasi					92%

Setelah filtering, tahap berikutnya adalah penggabungan data Kinect. Namun sebagai pembahasan awal, skeletal tracking untuk Kinect utama menghasilkan posisi joint dengan status TRACKED dengan tingkat akurasi hingga 92%, dimana penentuan status TRACKED mengikuti hasil skeletal tracking yang disediakan oleh Kinect SDK.

Tabel 5. Hasil Pengujian Data Gabungan

Live Performance	TOTAL	TRACKED	Penggantian	Peningkatan
Tubuh Atas + Konfigurasi 1	42020	38484	1	0%
Tubuh Bawah + Konfigurasi 1	37600	35003	280	0.58%
Tubuh Atas + Konfigurasi 2	41280	37586	151	0.29%
Tubuh Bawah + Konfigurasi 2	36960	34363	125	0.33%
Tubuh Atas + Konfigurasi 3	48240	42926	101	0.16%
Tubuh Bawah + Konfigurasi 3	34860	32853	2	0%
Tubuh Atas + Konfigurasi 4	39380	36179	140	0.35%
Tubuh Bawah + Konfigurasi 4	37420	35695	623	1.46%
Rata-rata Peningkatan				0.396%

Setelah data Kinect Utama diujikan, langkah berikutnya adalah pengujian test case dengan memanfaatkan data Kinect Pendukung. Data Kinect Pendukung yang digunakan dalam pengujian ini difilter terlebih dahulu sebelum digabungkan. data gabungan memberikan peningkatan akurasi hingga 0.396% terhadap akurasi skeletal tracking Kinect Utama. Rendahnya peningkatan akurasi, jika ditelaah, dapat disebabkan oleh dua faktor, yang

mana kedua faktor ini perlu diuji kembali kebenarannya dalam penelitian lebih lanjut, yaitu:

- Akurasi skeletal tracking untuk Kinect Utama sudah terlalu tinggi. Dengan tingginya akurasi ini, maka pemanfaatan Kinect Pendukung tidak terlihat memberikan peningkatan yang cukup signifikan.
- Test case terlalu ideal. Perlu diujikan test case dengan berbagai posisi yang berbeda untuk menunjukkan sumbangsih gabungan data. Salah satu kumpulan test case yang dapat digunakan adalah kumpulan gerakan yang motion capture yang dimiliki oleh Carnegie Mellon University, seperti terlihat pada Tabel 6.

Tabel 6. Test Case Carnegie Mellon University

Kategori	Subkategori
Interaction with Environment	Playground Uneven Terrain Path with Obstacles
Locomotion	Running Walking Jumping Varied
Physical Activities & Sports	Basketball Dance Gymnastics Acrobatics Martial Arts Racquet/Paddle Sports Soccer Boxing General Exercise & Stretching Golf Frisbee

Berdasarkan hasil uji coba untuk tiap pasang data, dengan konfigurasi yang dapat dilihat pada tabel 7, dapat disimpulkan bahwa model penggabungan dengan memanfaatkan 2 data Kinect memberikan peningkatan akurasi sebesar 3.91%. Data untuk konfigurasi dengan sudut 0° tidak berhasil, karena skeletal tracking tidak dapat memberikan hasil koordinat sendi.

Tabel 7. Konfigurasi Test Case CMU

Konfigurasi	Jumlah Rekaman
Sudut 90°	36 (depan dan kanan)
Sudut 45°	36 (depan dan kanan)
Sudut 0°	36 (atas dan bawah)
Total Rekaman	108

KESIMPULAN

Dari hasil penelitian tersebut, dapat disimpulkan beberapa hal sebagai berikut:

1. Status TRACKED yang diberikan oleh skeletal tracking Kinect SDK masih kurang valid untuk beberapa kasus. Pada beberapa bagian live performance, Kinect SDK menyatakan status

TRACKED untuk posisi sendi yang secara visual tidak wajar.

2. Kalman filter tradisional tidak dapat langsung diimplementasikan untuk smoothing data Kinect. Perlu modifikasi pada nilai gain agar Kalman Filter dapat digunakan untuk menjadi smoothing filter bagi data Kinect.
3. Filter Moving Average dapat digunakan untuk mengurangi jitter pada hasil Skeletal Tracking, namun hasil visual yang diberikan tidak sebaik Kalman Filter yang dimodifikasi. Namun untuk keperluan filtering sederhana, teknik ini dapat digunakan.
4. Teknik penggabungan dua data Kinect tidak memberikan kontribusi yang cukup signifikan dalam meningkatkan akurasi skeletal tracking. Hal ini dibuktikan dengan nilai peningkatan akurasi sebesar 3.91%, dimana total akurasi deteksi sendi dengan 2 Kinect adalah 82.93%.

Penelitian ini membuktikan bahwa filtering dan penggunaan Kinect kedua dapat sedikit meningkatkan akurasi skeletal tracking. Penelitian ini tidak berhenti di sini saja, namun dapat dikembangkan untuk penelitian selanjutnya, yaitu:

1. Hasil filtering dapat disempurnakan dengan memanfaatkan informasi depth yang disajikan oleh Kinect SDK. Nilai depth yang berada di sekitar joint dapat dimanfaatkan sebagai informasi tambahan untuk mengurangi jitter pada hasil skeletal tracking.
2. Sangat dimungkinkan penggunaan lebih dari 2 Kinect untuk mengumpulkan data skeletal tracking. Penelitian yang dapat dilakukan adalah menggunakan 4 Kinect yang diletakkan pada 4 penjuru mata angin dan menyorot model di tengah konfigurasi tersebut. Model penggabungan akan menentukan data dari Kinect manakah yang paling optimal, sehingga hasil skeletal tracking akan lebih baik.

DAFTAR PUSTAKA

- [1] OgranMotion, *Markerless Motion Capture*, <http://www.organicmotion.com/motion-capture/>, diakses pada 4 Agustus 2015.
- [2] Ashish Shingade, Archana Ghotkar, *Animation of 3D Human Model Using Markerless Motion Capture Applied To Sports*, International Journal of Computer Graphics & Animation (IJCGA) Vol.4, No.1, 2014.
- [3] Shotton et al., *Real-Time Human Pose Recognition in Parts from Single Depth Images*, 2010.
- [4] Jarrett Webb dan James Ashley, *Beginning Kinect Programming with the Microsoft Kinect SDK*, Apress, 2012.
- [5] Fletcher Dunn, Ian Parberry, *3D Math Primer for Graphics and Game Development*, Edisi Kedua, CRC Press, 2011.