

KOREKSI KEGAGALAN PELACAKAN UNTUK PELACAKAN OBJEK BERBASIS KERNEL

FAILURE TRACKING CORRECTION FOR KERNEL-BASED OBJECT TRACKING

Tembang Florian Falah¹, Suryo Adhi Wibowo, S.T., M.T. P.hD.²,
Raditiana Patmasari, S.T., M.T.³

^{1,2,3}Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

¹tembangflo@student.telkomuniversity.ac.id, ²suryoadhiwibowo@telkomuniversity.ac.id,

³raditiana@telkomuniversity.ac.id

Abstrak

Pelacakan objek merupakan salah satu bidang pada *computer vision* yang telah banyak diteliti dan semakin menunjukkan peningkatan yang semakin pesat dari tahun ke tahun, Dalam pengaplikasiannya pelacakan objek digunakan dalam melacak gerakan suatu benda maupun manusia hingga *augmented reality*. Meskipun sudah cukup canggih namun tetap saja akan mengalami kegagalan pelacakan, karena banyak faktor yang bisa menyebabkan gangguan pada objek dan dapat menyebabkan kegagalan dalam pelacakan.

Sistem ini akan bekerja dengan mendeteksi objek dalam suatu video dari setiap framenya, dengan menginisialisasi objek pada frame pertama lalu pelacakan di mulai dari *frame* kedua hingga *frame* terakhir dengan mengambil representasi *color histogram* dari objek yang telah diinisialisasi pada frame pertama, metode yang digunakan saat pelacakan adalah *mean-shift tracking*. Dalam pengaplikasiannya akan menggunakan algoritma *kernel based objek* untuk membangkitkan *mean-shift tracking*. Parameter pengujian untuk analisis kegagalan pelacakan yang digunakan dalam penelitian ini adalah dengan *bhattacharyya*, kita akan mengetahui seberapa besar sistem dapat melacak objek dengan benar dan mengetahui penyebab dari kegagalan pelacakan itu sendiri.

Hasil yang diperoleh dari penelitian ini adalah pengkoreksian kegagalan pelacakan dengan metode peningkatan performansi sistem. Parameter performansi pada tugas akhir ini menggunakan metode *precision plot* dan juga *success plot* dalam pengevaluasiannya dengan menggunakan *groundtruth* pada (OTB-50) sebagai dataset. Untuk *threshold bhattacharyya* menggunakan nilai 0,8 dan *image Enhancement* sebesar 1,4. Berdasarkan hasil percobaan yang dilakukan pada 5 buah *sequences* dari 50 *sequences* terpilih yaitu DragonBaby, Box, Basketball, CarDark, dan BlurFace. Didapatkan peningkatan performansi karena pengkoreksian terbesar sebesar 1,885 atau 1,88% pada *sequence* DragonBaby.

Kata kunci : Pelacakan objek, *mean-shift tracking*, *color histogram*, *tracking failure*, *bhattacharyya coefficient*.

Abstract

Object tracking is one of the fields in computer vision that has been widely investigated and increasingly shows an increasingly rapid increase from year to year. In its application object tracking is used in tracking the movements of objects and humans to augmented reality. Even though it is quite sophisticated, it will still experience tracking failure, because there are many factors that can cause interference to the object and can cause tracking failure.

This system will work by detecting objects in a video from each frame, by initializing the object in the first frame then tracking from the second frame to the last frame by taking a color histogram representation of the object initialized in the first frame, the method used when tracking is mean-shift tracking. In its application, it will use the kernel-based algorithm to generate mean-shift tracking. The test parameter for tracking failure analysis used in this study is with the Bhattachary, we will find out how much the system can track the object correctly and find out the cause of the tracking failure itself.

The results obtained from this study are correction of tracking failure with a system performance improvement method. The performance parameters in this final project use the precision plot method and also the success plot in evaluating it using groundtruth from (OTB-50) as a dataset. For the bhattachary threshold, use the value 0.8 and image enhancement of 1.4. Based on the results of experiments conducted on 5 sequences of 50 selected sequences, namely DragonBaby, Box, Basketball, CarDark, and BlurFace. Performance improvement was achieved because the biggest correction was 1,885 or 1.88% in the DragonBaby sequence.

Keywords: *object tracking, mean-shift tracking, color histogram, tracking failure, bhattacharyya coefficient.*

1. Pendahuluan

Pelacakan sebuah objek merupakan hal yang sangat diperlukan dalam berbagai bidang terutama dalam *computer vision* diberbagai aplikasi. Mulai dari pencitraan medis, *Closed Circuit Television* hingga untuk sistem keamanan dan masih banyak lagi. Banyak metode-metode penelitian yang digunakan untuk melakukan pelacakan objek. Namun, meskipun sudah banyak diteliti dan sudah mengalami peningkatan yang cukup signifikan, pelacakan objek masih saja memiliki masalah dikarenakan beberapa gangguan yang mengakibatkan kegagalan pelacakan. Berdasarkan *benchmark* gangguannya bisa meliputi: *occlusion, illumination variation, Scale Variation, Deformation, Motion Blur, Fast Motion, In-Plane Rotation, Out-of-Plane, Out-of-View, Background Clutters*, dan *low Resolution*[1]. Dengan adanya kegagalan pelacakan tersebut munculah berbagai penelitian untuk menemukan cara bagaimana agar dapat menjalankan pelacakan dengan baik dan tidak mengalami kegagalan (pengkoreksian).

Dalam penelitian ini penulis bermaksud ingin meneliti pelacakan objek dengan berbasiskan *kernel-based object tracking*. Sistem pelacakan berbasis *kernel-based object tracking* ini memiliki kelebihan lebih cepat dalam pemrosesannya[2], adapun metode yang digunakan merupakan *mean-shift* yang dimana metode ini berlaku pada citra berwarna pada video. Untuk pelacakan citra berwarna pada video, maka video tersebut harus dipresentasikan dalam bentuk distribusi histogram dari citra menggunakan distribusi histogram dari citra tersebut [3]. Pada saat pelacakan berlangsung sistem akan otomatis mendeteksi kegagalan pelacakan. Seperti yang sudah tak jarang pelacakan objek mengalami kegagalan pelacakan. Karena adanya kegagalan tersebut tentu akan mempengaruhi hasil dari *tracking target* oleh sistem menjadi tidak optimal. Sehingga, untuk mengatasi hal tersebut dilakukanlah pengkoreksian kegagalan pelacakan yang dilakukan dengan memanfaatkan histogram pada *frame* sebelum kegagalan terjadi. Selain itu, penelitian ini pada kernel digunakan *threshold* dengan nilai 0,7 untuk objek dianggap hilang atau gagal pelacakan, objek akan dianggap hilang atau tidak terdeteksi saat sistem yang berjalan mengeluarkan nilai kurang dari 0,7. Selain itu digunakan juga metode *bhattacharyya coefficient* untuk mengetahui kesimilaritasan suatu gambar berdasarkan distribusi warna, rentang nilai dari *bhattacharyya coefficient* dalam penelitian ini adalah 0 – 1, dimana semakin mendekati 1 semakin mirip objek pada *frame* sekarang dengan objek pada *frame* sebelumnya. *Threshold* yang digunakan pada *bhattacharyya* adalah 0,8 dan jika pelacakan mendeteksi objek dan bernilai kurang dari *threshold* yaitu 0,8 maka akan terjadi kegagalan pelacakan.

Hasil dari penelitian ini bertujuan untuk mendapatkan performansi dari sistem yang dilacak dan performansi sistem yang mengalami pengkoreksian kegagalan pelacakan. Dengan pemanfaatan pengkoreksian kegagalan pelacakan ini diharapkan keluaran sistem akan lebih baik. Selain daripada itu akan diteliti juga penyebab-penyebab dari sistem gagal dalam melakukan pelacakan sebuah objek.

2. Dasar Teori

2.1 Video Digital

Video merupakan penyimpanan sekumpulan gambar bergerak yang dapat ditampilkan pada layar komputer dan mengacu pada hasil dari proses pengambilan[4]. Video pada dasarnya merupakan *array* tiga dimensi. Dua dimensi digunakan untuk menggambarkan ruang pergerakan citra (*spasial*), dan satu dimensi lainnya menggambarkan waktu. Video digital mempunyai beberapa keunggulan jika dibandingkan dengan video analog, seperti dalam proses memanipulasi lebih mudah, dapat disimpan dalam jangka waktu yang sangat panjang dan juga aksesibilitasnya karena video digital dapat disebar luaskan melalui internet. Video digital tersusun atas serangkaian *frame* yang ditampilkan dalam durasi waktu tertentu sehingga dikenal dengan *fps (frame per second)*. Atribut-atribut dalam video digital adalah sebagai berikut:

2.2 Citra Digital [5]

Secara umum, citra digital merupakan citra yang dapat diolah oleh komputer. Pengolahan citra digital diperlihatkan pada pemrosesan gambar dua dimensi. Citra digital juga merupakan sebuah *array* yang berisi nilai-nilai *real* dan kompleks yang dipresentasikan berupa deretan bit-bit tertentu.

Citra digital tersusun atas sejumlah elemen yang masing-masing memiliki lokasi dan nilai/intensitas tertentu. Elemen-elemen tersebut bisa berupa elemen gambar, elemen citra, elemen *pels*, dan juga elemen piksel.

2.3 Pelacakan Objek

Pelacakan objek yang banyak diteliti sebelumnya biasanya terbagi menjadi dua bagian, yaitu *generative* dan *discriminative*. Dimana *generative* merupakan metode yang menggunakan *generative models* lalu menyesuaikan area target dengan model terbaik yang sesuai. Sementara *discriminative* adalah sebuah metode

yang bertujuan membuat model dapat membedakan antara objek dan *background* [6]. Dalam pelacakan objek sendiri terdapat beberapa gangguan yang membuat proses pelacakan objek sulit dilakukan, berdasarkan *benchmark* gangguannya meliputi: *occlusion, illumination variation, Scale Variation, Deformation, Motion Blur, Fast Motion, In-Plane Rotation, Out-of-Plane, Out-of-View, Background Clutters*, serta *low Resolution*[1]. Objek merupakan sebuah konsep, abstraksi atau merupakan sesuatu yang diberi batasan jelas dan dimaksudkan untuk sebuah aplikasi sebuah objek adalah sesuatu yang mempunyai keadaan, perilaku dan identitas. Keadaan dari objek adalah satu dari kondisi yang memungkinkan dimana objek dapat muncul dan dapat secara normal berubah berdasarkan waktu.

2.4 Mean-Shift [7]

Mean shift merupakan sebuah metode pelacakan objek berbasis *density* (kepadatan) sebuah ciri dan juga bisa sebagai alternatif untuk pelacakan pemfilteran partikel [8]. *Mean shift* merupakan algoritma *non-parametrik* (tidak memerlukan persamaan untuk mendistribusi). *Mean shift* yang digunakan pada penelitian ini yaitu dengan menggunakan *color histogram* atau warna sebagai acuan saat *tracking*. Dalam banyak aplikasi *mean shift* dapat melakukan *tracking* pada objek *non rigid* atau objek yang berubah dan tidak tajam. *Non rigid* di deskripsikan dengan *color histogram* sehingga objek akan tetap terdeteksi walau ukuran, lokasi, bahkan bentuk sedikit berubah sekalipun.

2.5 Kernel Based-Object Tracking

Kernel merupakan sebuah metode yang sudah terbukti sukses diimplementasikan pada sejumlah metode *Learning* seperti *Gaussian Process, Regularization Network*, dan SVM [9]. Selain daripada itu penggunaan fungsi kernel sebagai pengukur kepadatan merupakan metode dalam pemrosesan gambar yang menarik banyak perhatian, pelacakan dengan metode kernel juga merupakan pelacakan dengan teknik *nonparametrik*[10]. Pada dasarnya metode ini merupakan algoritma detektor mode dalam ruang distribusi kepadatan, dalam metode ini menetapkan bobot pada piksel dalam jendela.

2.4.1 Target Model

Sebuah target model akan diwakili oleh daerah persegi pada bingkai video frame pertama dan wilayah ini disebut juga *region of interest* (ROI) dinormalisasi sebagai wilayah target. Dalam pengaplikasiannya kami memilih warna sebagai fitur dalam model target dan wilayah calon target. Karena kita harus mengukur kesamaan antara dua wilayah terpilih dalam dua bingkai berturut-turut melalui kriteria pengukuran kesamaan *bhattacharyya*. Piksel pada daerah persegi panjang diberi bobot sesuai dengan jaraknya dari pusat lingkaran normal yang dikelilingi persegi panjang tersebut.

2.4.2 Target Kandidat

Target kandidat dalam *frame* saat ini bertujuan untuk penentuan wilayah yang dimungkinkan suatu target pelacakan berada. Perhitungan bobot target yang mungkin menjadi wilayah target sasaran dalam lingkup R-G-B. Pada proses ini wilayah yang akan dihitung adalah seluruh wilayah target kandidat yang mungkin menjadi lokasi pusat wilayah target. Selain daripada itu terdapat juga bobot vektor yang dimana digunakan untuk mencari objek target sasaran.

2.4.3 Color Histogram

Probability Density Function (PDF) dari warna u di representasikan dengan menggunakan $m - \text{bin}$ *color histogram* dan ruang warna (R, G, B) dibagi menjadi $16 \times 16 \times 16$ bin [8]. Setiap bin menunjukkan jarak dari nilai piksel, misalnya bin0 merupakan (0 – 15), jarak dari bin1 adalah (16 – 31) dan seterusnya hingga bin15 merupakan (240 – 255). Nilai dari m adalah 4095 warna dan jarak dari warna adalah u yang bernilai 0 sampai 4095 (m).

2.4.4 Lokalisasi Target

Lokalisasi target merupakan penentuan dari mana letak objek target berada. Untuk mendapatkan lokalisasi yang berhubungan dengan target pada frame saat ini, maka jarak target harus diminimalisasi terlebih dahulu dengan memaksimalkan nilai dari koefisien *Bhattacharyya* [10]. Prosedur lokalisasi dimulai dari posisi target pada frame sebelumnya atau target model lalu dicari pada daerah sekitarnya menggunakan informasi *gradient* yang di hasilkan oleh vektor *mean - shift* yang merupakan hasil kepadatan nilai piksel target dan di dapat dari vektor *mean-shift*.

2.5 Bhattacharyya Coefficient

Fungsi dari *Bhattacharyya* sendiri digunakan untuk menentukan jarak antara target model dan target kandidat [10]. Perhitungan dari *Bhattacharyya* sendiri dilakukan setelah lokasi target kandidat objek yang telah mirip dengan target objek ditemukan. Metode ini digunakan untuk mengetahui kesimilaritasan suatu gambar berdasarkan distribusi warna [10]. Hasil keluaran dari koefisien *bhattacharyya* merupakan nilai yang sudah di tentukan, dimana semakin mendekati nilai terbesar dari angka yang sudah di tentukan maka semakin mirip antara target model dan target kandidat.

2.6 Object Tracking Benchmark (OTB-50)

Object Tracking Benchmark (OTB-50) dataset memiliki kualitas yang baik dan biasa digunakan untuk *in-depth analysis*. Pada dasarnya OTB-50 merupakan 50 *sequences* tersulit dari 100 *sequences* yang ada pada

Object Tracking Benchmark (OTB-100).

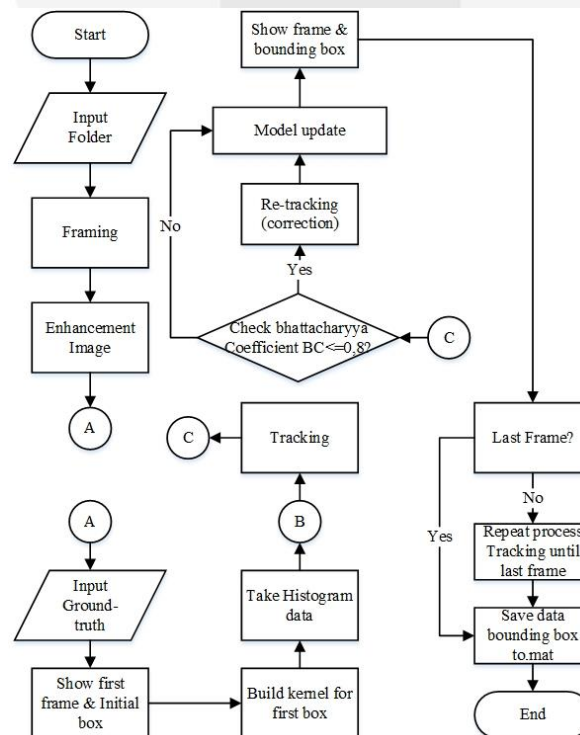
Tabel 1 Penjelasan Data Atrinut OTB-50 Benchmark Dataset.

Nama	Deskripsi
OCC	<i>Occlusion</i> adalah kondisi dimana keberadaan objek terhalangi sebagian atau seluruhnya.
IL	<i>illumination variation</i> adalah perubahan cahaya pada objek yang cukup besar
SV	<i>Scale Variation</i> adalah rasio keluaran nilai di atas patas pada <i>bounding box</i> pada bingkai pertama dan bingkai yang sedang dijalankan.
DEF	<i>Deformation</i> adalah pengolahan citra dengan <i>non-rigid deformation</i> .
MB	<i>Motion Blur</i> merupakan wilayah dimana target kabur karena gerakan target atau gerakan kamera.
FM	<i>Fast Motion</i> adalah gerak dari <i>ground-truth</i> lebih besar dari tm piksel ($tm = 20$).
IPR	<i>In-Plane Rotation</i> adalah rotasi objek ke dalam pada <i>image plane</i> .
OPR	<i>Out-of-Plane</i> adalah rotasi objek ke luar pada <i>image plane</i> .
OV	<i>Out-of-View</i> adalah dimana beberapa objek meninggalkan bingkai citra.
BC	<i>Background Clutters</i> merupakan keadaan dimana warna atau tekstur target mirip dengan background.
LR	<i>Low Resolution</i> dimana jumlah piksel di dalam <i>ground-truth bounding box</i> kurang dari tr ($tr = 400$).

3. Perancangan Sistem

3.1. Diagram Blok Secara Umum

Dalam tugas akhir ini dibuat permodelan sistem deteksi objek dengan data masukan berupa video dari *Object Tracking Benchmark (OTB-50)*. Secara umum penelitian ini memiliki skema sebagai berikut:



Gambar 1 Diagram sistem.

3.2 Parameter Performansi

Parameter performansi merupakan pengujian algoritma pelacakan objek dengan parameter - parameter yang telah ditentukan. Adapun parameter – parameter yang digunakan adalah *precision plot* dan *success plot* pada metode *Area Under Curve*.

a. Precision plot

Digunakan untuk pengukuran rata-rata *euclidean* antara titik tengah lokasi pencarian objek oleh sistem. Ketika algoritma pelacakan objek kehilangan target, maka lokasi target secara langsung akan menjadi acak, dan rata-rata error tidak dapat mengukur performansi pelacakan objek dengan benar[1].

b. Success plot

Merupakan pengujian yang menggunakan nilai irisan [10]. Pada lokasi *bounding box* objek M dan lokasi *ground-truth* B bertumpuk maka nilai irisan dapat didefinisikan:

$$s = \frac{M \cap B}{M \cup B}$$

Dimana \cap merepresentasikan irisan, \cup merepresentasikan gabungan, dan $|\cdot|$ merepresentasikan jumlah piksel. Nilai rata-rata dari irisan (AOS) dapat dijadikan sebagai pengukuran performansi. Nilai irisan juga dapat menentukan apakah suatu algoritma berhasil melakukan pelacakan objek dalam satu frame, dengan melakukan pengecekan nilai s lebih besar dari nilai batas *ground-truth*.

4. Hasil dan Analisis

4.1 Tahapan Pengujian sistem

1. Tahap pertama

Melakukan pemilihan satu sequence dari jumlah 5 sequence video terpilih dengan kualitas video yang berbeda, baik dari segi kualitas citra, pergerakan objek, kombinasi warna, dan banyaknya objek dalam satu frame.

2. Tahap Kedua

Melakukan preprocessing, dimana pada proses ini citra pada seluruh sequence akan dilakukan image enhancement berupa dengan mengubah-ubah contrast. Hal ini dilakukan untuk mencari nilai parameter pada code yang terbaik.

3. Tahap Ketiga

Setelah sistem dijalankan dengan menggunakan salah satu sequence dari *Object Tracking Benchmark* (OTB-50) lalu dilakukan inialisasi objek dan mengambil nilai *ground-truth bounding box* pada setiap sekuen data.

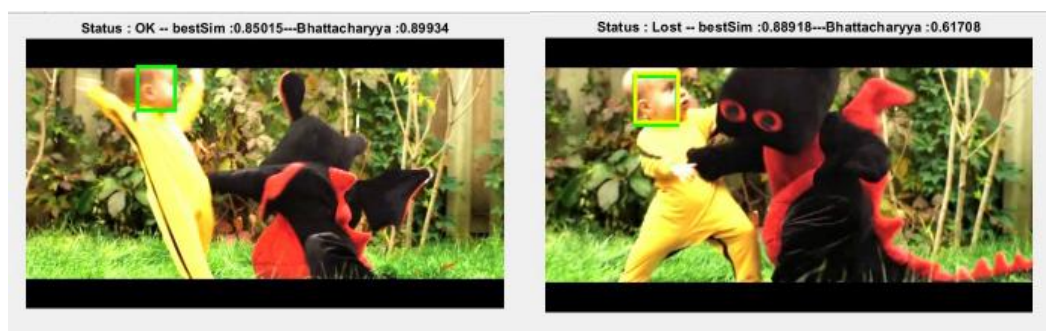
4. Tahap Keempat

Setelah seluruh sekuen data berjalan dengan lancar dan berhasil mencapai *frame* terakhir, nilai dari *bounding box* dari setiap iterasi disimpan di *database* yang telah disediakan yang nantinya akan berbentuk .mat. Penyimpanan nilai *boundi box* terbagi atas dua *database* yaitu *database* nilai *bounding box before correction* dan *database* nilai *bounding box after correction*.

4.2 Hasil Pengujian sistem dan Analisis terhadap Groundtruth

4.2.1 Success Plot

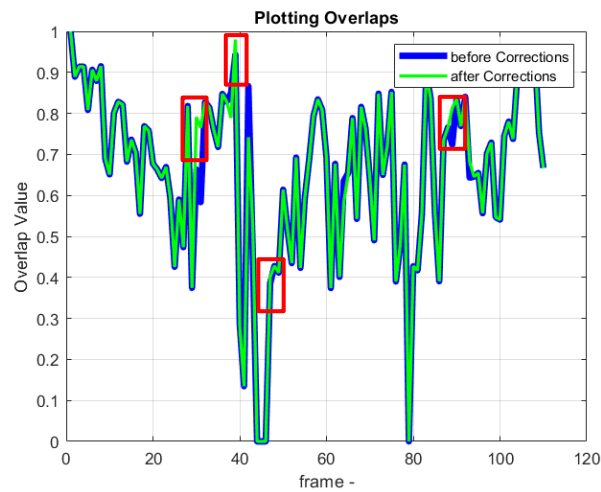
Adapun pengujian dari sisitem adalah dengan menggunakan video DragonBaby dari OTB-50.



Gambar 2 Bounding box pada video DragonBaby.

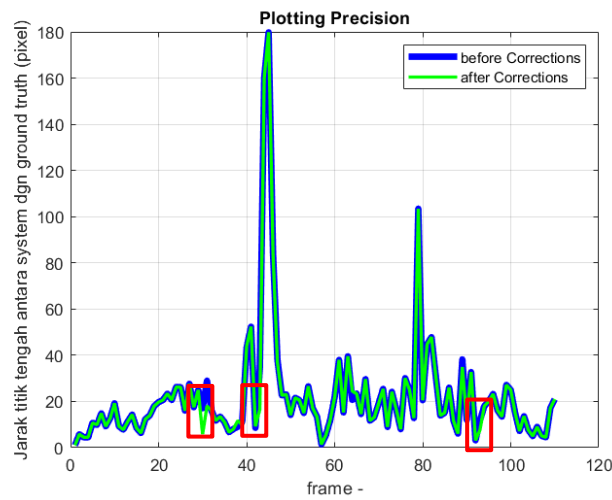
Telihat pada saat pelacakan status sistem pada *frame* kiri terdeteksi OK dan nilai *bhattacharyya* = 0,89934 > 0,8. Sementara pada *frame* sebelah kiri terpantau terjadi pengkoreksian karena nilai *bhattacharyya* = 0,61708 < 0,8. Nilai ini kurang dari nilai *threshold* yang sudah di tetapkan. Terdapat atribut SV, OCC, MB, FM, IPR,

OPR, dan OV dalam video DragonBaby.



Gambar 3 Success Plot pada video DragonBaby.

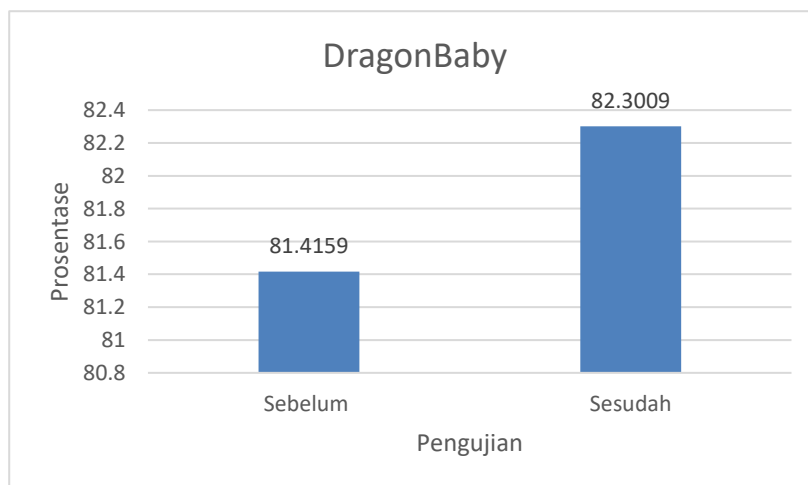
4.2.2 Precision Plot



Gambar 4 Precision Plot pada video DragonBaby.

Dapat dilihat dari Gambar 4, Precision plot pada sequence DragonBaby, dimana semakin mendekati 0 atau bahkan 0 maka akan semakin baik sistem itu. Dapat dilihat dari frame ke 30, frame 43, dan frame 95 terdapat pengkoreksian yang membuat grafik makin turun ke bawah mendekati 0.

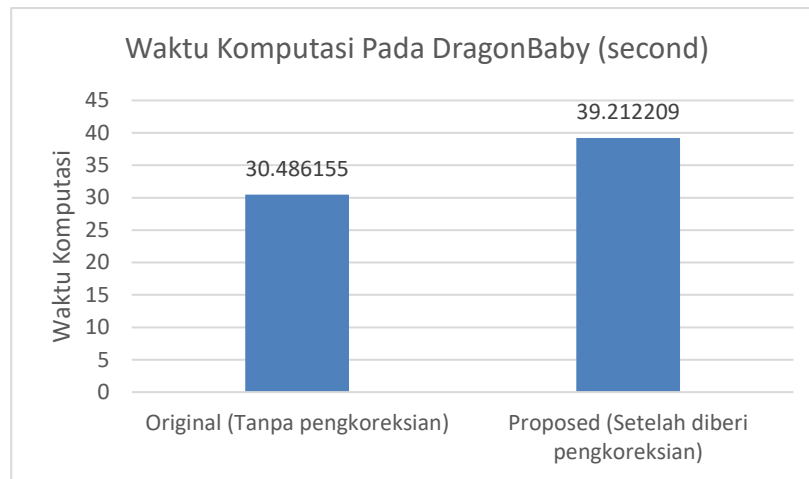
4.2.3 Analisis Performansi Sistem



Gambar 5 Perbandingan performansi pada sistem original dan sistem proposed.

Dapat dilihat dari grafik perbandingan performansi antara sistem original dan sistem proposed (sesudah pengkoreksian) performansi/akurasi pada pelacakan objek mengalami kenaikan sebesar 1,885 atau jika di prosentase naik sebesar 1,88%.

4.2.4 Waktu Komputasi



Gambar 6 Selisih waktu komputasi original dan proposed

Dapat disimpulkan dari waktu komputasi pada sequence DragonBaby dengan jumlah frame sebanyak 113 frames sistem original mendapatkan waktu 30.486155s dengan waktu komputasi 0.269788s/frame sementara setelah dijalankan pengkoreksian (proposed) adalah 39.212209s dengan waktu komputasi 0.347010s/frame. Selisih waktu dari kedua sistem yang dijalankan pada sequence DragonBaby ini adalah 8.726054s.

5. Kesimpulan

Setelah dibuatnya sistem dan dilakukan analisis pada sistem tersebut dengan berdasarkan pada parameter-parameter yang telah ditentukan, maka dapat diambil kesimpulan sebagai berikut:

1. Untuk menaikkan performansi pada pelacakan dapat dilakukan dengan *re-tracking*, hasil yang di dapatkan adalah koreksi kegagalan pelacakan berhasil dilakukan dengan *proposed (re-tracking)* terdapat peningkatan sebesar 1,885% pada DragonBaby, 0,7632% pada CarDark, 0,1721% pada Box, 0,2029% pada blurFace, dan 0,8276% pada Basketball.
2. Algoritma *original* kernel mengalami kegagalan karena berbagai macam kendala dan gangguan, berdasarkan *benchmark* gangguannya meliputi: *occlusion, illumination variation, Scale Variation, Deformation, Motion Blur, Fast Motion, In-Plane Rotation, Out-of-Plane, Out-of-View, Background Clutters*, serta *low Resolution*, serta dapat dilihat kelemahan kernel adalah tidak dapat mendeteksi objek dengan tepat.
3. Pada *original method sequence* DragonBaby tercatat membutuhkan waktu 0.269788s/frame sementara pada *proposed* 0.347010s/frame dengan kenaikan 0,077222s/frame. *Sequence* CarDark pada *original method* 0.108338s/frame sementara pada *proposed* 0.109768s/frame kenaikan sebesar 0,00143s/frame. *Original method sequence* Box 0.149407s/frame *proposed method* 0.157244s/frame dengan kenaikan 0.00783s/frame. BlurFace pada *original method* 0.169994s/frame sementara pada *proposed* 0.188666s/frame kenaikan waktu adalah 0,01867s/frame. Basketball *original method* 0.124108s/frame sementara pada *proposed method* 0.129841s/frame dengan kenaikan 0,00573s/frame. Waktu pada pelacakan *original* dan pelacakan *proposed* berbeda, pelacakan *proposed* memakan waktu lebih lama dikarenakan ada proses *tracking* yang dimana sistem berarti bekerja lebih.
4. Pada pengujian *threshold* digunakan nilai *bhattacharyya* 0.8 dan *image enhancement* 1.4, hal ini karena berdasarkan berkali-kali pengujian nilai tersebut cocok dengan hampir semua *sequence* yang ada di OTB-50 benchmark.

Daftar Pustaka:

- [1] Y. Wu, J. Lim, M. H. Yang, "Object Tracking Benchmark" IEEE Trans. Pattern Anal. Mach Intell., vol. 37, no. 9, pp. 1834 – 1848, 2015.
- [2] Wilson, Liliana, G. Kartika, "Perancangan dan Pembuatan Aplikasi Tracking Object Pada Video dengan Kernel-Based"
- [3] https://docs.opencv.org/3.1.0/db/df8/tutorial_py_meanshift.html (diakses pada tanggal 21 Desember 2018 jam 0:15)
- [4] M. M. Dehshibi, A. Vaafanezhad, J. Shanbehzadeh, "Kernel-based Object Tracking using Particle Filter

- with Incremental Bhattacharyya Similarity” IEEE International Conference on Hybrid Intelligent Systems, pp. 49 – 53, 2013.
- [5] Y. Awatsuji, P. Xia, O. Matoba, “One-mega-frame-per-second phase-shifting digital holography”, IEEE 13th International Conference on Industrial Informatics (INDIN), pp. 721 – 726, Cambridge, UK, 2015.
- [6] H. Nam, B. Han, “Learning Multi-Domain Convolutional Neural Networks for Visual Tracking”, *arXiv Prepr. arXiv1510.07945*, pp. 4293 – 81 – 106, 1986.
- [7] W. Lei, L. Yixiong, C. Wangyang, Z. Bei, “Failure Detection and Correction for Appearance Based Facial Tracking” Chinese Journal of Electronics, pp. 20 – 25, vol.24 , no.1, January 2015.
- [8] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-based object tracking,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 5, pp. 564–577, 2003.
- [9] S. A. Wibowo, H. Lee, E. K. Kim, T. Kwon, S. Kim, “Tracking Failures Detection and Correction for Face Tracking by Detection Approach Based on Fuzzy Coding Histogram and Point Representation” in Proceedings of 2015 International Conference on Fuzzy Theory and Its Applications (iFUZZY), pp. 34 - 39, Yilan, Taiwan, November 2015.
- [10] L. Zhang, D. Zhang, Y. Su, F. Long, “Adaptive Kernel-bandwidth Object Tracking Based on Mean-shift Algorithm” IEEE Fourth International Conference on Intelligent Control and Information Processing (ICICIP), pp. 413 - 416 , Beijing, China, September 2013.
- [11] H. An-qi, H. Zhi-qiang, Y. Wang-sheng, L. Xiang, “A Visual Object Tracking Method Based on Improved Bhattacharyya Coefficient and Model Update Strategy”, IEEE Fifth International Conference on Intelligent Systems Design and Engineering Application, pp. 100 - 105, 2014.

