

## PENINGKATAN PERFORMANSI KERNEL-BASED OBJECT TRACKING MENGUNAKAN TYPE-2 FUZZY LOGIC

### PERFORMANCE IMPROVEMENT OF KERNEL-BASED OBJECT TRACKING USING TYPE-2 FUZZY LOGIC

Muhammad Alief Hidayah Baso<sup>1</sup>, Suryo Adhi Wibowo, S.T., M.T. P.hD.<sup>2</sup>,  
Ledy Novamizanti, S.Si., M.T.<sup>3</sup>

<sup>1,2,3</sup>Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

<sup>1</sup>[aliefhidayah@student.telkomuniversity.ac.id](mailto:aliefhidayah@student.telkomuniversity.ac.id), <sup>2</sup>[suryoadhiwibowo@telkomuniversity.co.id](mailto:suryoadhiwibowo@telkomuniversity.co.id),

<sup>3</sup>[ledyaldn@telkomuniversity.ac.id](mailto:ledyaldn@telkomuniversity.ac.id)

#### Abstrak

*Object tracking* merupakan salah satu bidang pada *computer vision* yang mempelajari tentang cara melacak suatu objek yang bergerak pada suatu ruang, yang dimana sekarang sedang berkembang dengan pesat. *Object tracking* dalam pengaplikasiannya digunakan dalam melacak gerakan benda maupun manusia dan *augmented reality*. Namun dalam pengaplikasiannya *object tracking* masih memiliki banyak tantangan dalam mendeteksi suatu objek. Pada *kernel-based object tracking*, kemiripan warna (*background clutters*) dan pergerakan cepat dari objek (*motion blur*) merupakan faktor utama penyebab terjadinya kegagalan pelacakan dan juga pada *kernel-based object tracking* belum dapat mendeteksi kegagalan dalam pelacakan. Sehingga pada algoritma koreksi kegagalan diterapkan pada *kernel-based* menggunakan *type-2 fuzzy logic*.

*Kernel-based object tracking* merupakan suatu metode pelacakan objek yang memprediksi lokasi objek berdasarkan histogram warna yang memiliki probabilitas tertinggi. Dalam pengaplikasiannya metode *kernel-based* kurang akurat dalam mendeteksi objek dikarenakan kemiripan warna *background* dengan objek yang di *tracking* serta *tracker* tetap akan melakukan *tracking* dengan prediksi yang salah sehingga digunakan *type-2 fuzzy logic* untuk menganalisis terjadinya kegagalan sesuai dengan *rules* yang telah ditentukan sebelumnya dan melakukan koreksi pada saat *object tracking*. Dimana *output* dari metode *kernel-based* menjadi *input* dari *type-2 fuzzy logic*.

Hasil dari algoritma sistem yang dibuat ini dapat meningkatkan kinerja dari metode *kernel-based* dari algoritma diuji menggunakan *Object Tracking Benchmark 50 (OTB-50)* berdasarkan parameter *precision plot* dan *success plot*. Dari hasil percobaan yang telah dilakukan, algoritma yang diusulkan menghasilkan performa sebesar 0.001 lebih presisi dibandingkan dengan *kernel-based object tracking* berdasarkan parameter *precision plot* dan memiliki tingkat kesuksesan lebih besar 0.003 diukur berdasarkan parameter *success plot*.

**Kata kunci :** *Object Tracking, Kernel-based, Type-2 Fuzzy Logic, Computer Vision*

#### Abstract

*Object tracking* is one area in *computer vision* that learns about how to track an object that moves in a space, which is now growing rapidly. *Object tracking* in application is used in tracking both object movements and humans and *augmented reality*. But in application *object tracking* still has many challenges in detecting an object. In the *kernel-based object tracking*, color similarity (*background clutters*) and fast motion of objects (*motion blur*) are the main factors causing tracking failure and also on the *kernel-based object tracking* that have not been able to detect failures in training. So that the failure correction algorithm is applied to *kernel-based* using *type-2 fuzzy logic*.

*kernel-based object tracking* is a method of tracking objects that predicts the location of an object based on a color histogram that has the highest probability. In its application the *kernel-based* method is less accurate in detecting objects because the similarity of background colors with objects that are tracked and *tracker* will still track incorrect predictions so that *Type-2 fuzzy logic* is used to analyze the occurrence of failure according to predetermined rules and do correction when *object tracking*. Where the output of the *kernel-based* method is an input of *type-2 fuzzy logic*.

The results of this system algorithm can improve the performance of the *kernel based* method of the algorithm tested using a *Object Tracking Benchmark 50 (OTB50)* based on the parameters of the *Precision Plot* and *Success Plot*. From the results of experiments that have been done, the proposed algorithm produces a performance of 0.001 more precise than *kernel-based object tracking* based on *precision plot* parameters and has a success rate greater than 0.003 measured based on *success plot* parameter.

**Keywords:** *Object Tracking, Kernel-based, Type-2 Fuzzy Logic, Computer Vision*

## 1. Pendahuluan

*Object tracking* adalah salah satu bidang dari *computer vision* dimana sistem akan bekerja melacak suatu objek yang bergerak dalam suatu sekuen video. *Object tracking* banyak digunakan dalam pengawasan dalam suatu ruangan, kompresi video, pembantu dalam mengemudi, dan *augmented reality*.

*Object tracking* dalam pengaplikasiannya menggunakan metode antara lain *template matching*[1], *Lucas Kanade*[2], *contour tracking*[3] dan sebagainya yang memiliki kelebihan dan kekurangannya masing-masing dibandingkan *kernel-based*, tetapi metode *kernel-based* memiliki kelebihan dimana prosesnya yang cepat akan tetapi kurang akurat dalam mendeteksi objek[4]. Yang menjadikan performansi *kernel-based* kurang baik.

*Kernel-based object tracking* adalah pelacak suatu gerakan dalam suatu objek dalam video yang kinerjanya berbasis histogram warna sesuai dengan *initial state* yang telah kita berikan di *frame* pertama dari video[5]. Jika objek dalam video terdeteksi maka suatu objek bergerak dengan latar belakangnya akan dipisah dalam video tersebut. Dalam kinerjanya *kernel-based* memiliki proses yang cepat namun kurang akurat dalam mendeteksi objek[6]. Pada sistem yang dibuat ini kinerja dari *kernel-based* akan ditingkatkan menggunakan *type-2 fuzzy logic*. Dimana akan menjadi *input* dari *output kernel-based* yang akan mengatasi ketidaktepatan (*imprecise*)[7] dan melakukan koreksi terhadap *output* dari *kernel-based*[8]. *Type-2 fuzzy logic* adalah peningkatan performa kinerja dari *type-1 fuzzy logic* yang merupakan suatu sistem yang membuat kecerdasan buatan yang kinerjanya peningkatan dari logika boolean yang hanya dapat diekspresikan dalam istilah biner 0 dan 1 menjadi memungkinkan nilai keanggotaan antara 0 dan 1 dalam bentuk linguistik, konsep tidak pasti seperti "sedikit", "lumayan", dan "sangat". *Type-2 fuzzy logic* diaplikasikan pada robot[9] yang memerlukan kecerdasan buatan, koreksi terhadap pelacakan[10] dan sebagainya.

Pada sistem yang dibuat telah ditingkatkan performansi dari *kernel-based object tracking* dengan menggunakan *type-2 fuzzy logic*, dimana objek yang dilacak merupakan *benchmark dataset* OTB-50 yang merupakan sebuah sekuen video yang dimana videonya bukan *real-time*. Hasil dari pelacakan *kernel-based* akan dikoreksi oleh *type-2 fuzzy logic* sesuai dengan parameter yang ditentukan sehingga objek yang dilacak dapat lebih tepat sehingga meningkatkan performansi *kernel-based object tracking* dengan parameter *success plot* dan *precision plot* dengan metode *One Pass Evaluation* (OPE)[11].

## 2. Dasar Teori

### 2.1. Video

Video merupakan sekumpulan gambar yang tersusun dalam urutan tertentu dan diputar dengan kecepatan tertentu sehingga dihasilkan gambar yang seolah-olah bergerak. Biasanya penyusun video disebut dengan *frame* video atau sekuen video. Sebuah video digital disusun oleh kumpulan citra digital yang dijalankan dengan suatu kecepatan tertentu (*frame/second*).

#### 2.1.1. Pengolahan Video Digital

Pengolahan video digital, objek yang diolah sesungguhnya adalah semua citra digital penyusun suatu video sehingga pengolahan data yang terjadi pada simulasi ini yaitu pengolahan citra digital yaitu berupa *frame video*. Pengolahan video itu sendiri adalah proses manipulasi citra digital dengan bantuan computer.

### 2.2. Object Tracking Benchmark 50

*Object Tracking Benchmark 50* merupakan dataset turunan dari *ObjectTracking Benchmark 100* (OTB-100). *Object Tracking Benchmark 50* (OTB-50) terdiri dari 50 objek target pilihan dari OTB-100 yang memiliki tingkat kesulitan tinggi dan sering digunakan untuk *in-depth analysis*. Adapula beberapa sekuen memiliki kondisi video yang terdiri *frame* yang memiliki format warna *greyscale* dan juga RGB. Terdapat 11 atribut yang digunakan untuk menganalisis kekuatan dan kelemahan suatu algoritma pelacakan seperti yang ditunjukkan pada tabel 1. Setiap atribut memiliki faktor tantangan khusus pada *object tracking* dengan setiap *sequence* pada OTB-50 terdiri dari beberapa atribut berbeda[11].

**Tabel 1** Penjelasan Data Atribut OTB-50 *Benchmark Dataset*.

Nama	Deskripsi
OCC	<i>Occulusion</i> adalah kondisi dimana keberadaan objek terhalangi sebagian atau seluruhnya.
IL	<i>ilumination variation</i> adalah perubahan cahaya pada objek yang cukup besar
SV	<i>Scale Variation - ratio</i> antara bounding box pada frame pertama dan frame selanjutnya berbeda
DEF	<i>Deformation</i> adalah pengolahan citra dengan <i>non-rigid</i>

	<i>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 ( $t_m = 20$ ).
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 <i>background</i>
LR	<i>Low Resolution</i> dimana jumlah piksel di dalam <i>ground-truth bounding box</i> kurang dari $t_r$ ( $t_r = 400$ ).

### 2.3. Kernel-based Object Tracking

*Kernel-based object tracking* merupakan salah satu metode pelacakan objek dimana pada suatu video digital yang didalam tiap *frame*-nya akan ditentukan suatu objek yang akan dilacak dimana langkah-langkah pelacakan dimulai dari menentukan target model lalu target kandidat menggunakan pencarian warna RGB[13] yang akan ditentukan sebagai objek yang akan dilacak, *bhattacharya coefficient*, lokalisasi target dan minimalisasi jarak[5].

#### 2.3.1. Target Model

Sebuah objek yang akan dilacak diwakili oleh daerah elipsoidal, untuk menghilangkan pengaruh dari perbedaan target dimensi semua target dinormalkan menjadi satuan lingkaran. Ini dapat diperoleh dengan cara rescaling secara independen pada dimensi baris dan kolom dengan  $h_x$  dan  $h_y$ .

Dimana variabel  $\{x_i\}_{i=1...n}$  menjadi lokasi *pixel* yang telah di normalisasikan pada daerah yang akan dijadikan target model dimana daerah pusat dari target model berada pada 0. sebuah kernel isotropik, dengan penurunan kernel cembung dan monoton memberikan bobot lebih kecil pada *pixel* yang jauh dari pusat. Dengan menggunakan kenaikan bobot ini kekokohan dari estimasi kerapatan *pixel* paling dapat diandalkan, yang sering dipengaruhi oleh oklusi atau gangguan dari *background*.

#### 2.3.2. Target Kandidat

Target kandidat dalam *frame* saat ini bertujuan untuk penentuan wilayah yang dimungkinkan suatu target pelacakan berada. Perhitungan bobot target yang mungkin mejadi wilayah target sasaran dalam lingkup RGB. 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.

#### 2.3.3. Color Histogram

*Probability density function* (PDF) menggunakan histogram warna yang merupakan representasi grafis untuk distribusi warna dari citra digital, dari warna pada sebuah *frame* direpresentasikan dengan menggunakan nilai *pixel* untuk menentukan *color histogram* dan ruang warna (R, G, B) pada *frame* tersebut. Seperti misal, setiap bin menunjukkan jarak dari nilai *pixel*, misalnya bin 0 adalah (0-15) nilai range warna dari *pixel*-nya, jarak dari bin 1 adalah (16 - 31) dan seterusnya hingga bin 15 adalah (240 - 255) dimana nilai tersebut merepresentasikan warna yang terdapat pada bin yang terdapat pada *pixel* dari *frame* tersebut.

#### 2.3.4. Bhattacharyya Coefficient

Fungsi dari *Bhattacharyya* sendiri digunakan untuk menentukan jarak antara target model dan target kandidat. Perhitungan dari *Bhattacharyya* sendiri dilakukan setelah lokasi target kandidat objek yang telah mirip dengan target objek ditemukan[14]. Metode ini digunakan untuk mengetahui kesimilaritasan suatu gambar berdasarkan distribusi warna. 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.

#### 2.3.5. Lokalisasi Target

Untuk mendapatkan lokasi yang berhubungan dengan target pada *frame* saat ini, jarak target harus diminimalisasi dengan cara memaksimalkan nilai dari *bhattacharyya*. Prosedur lokalisasi dimulai dari posisi target pada *frame* sebelumnya dan dicari pada daerah sekitarnya menggunakan informasi gradien yang dihasilkan oleh vektor *mean shift*[15] yang mencari kepadatan dari nilai *pixel* yang menjadi target.

### 2.3.6. Minimalisasi Jarak

Meminimalkan jarak sama dengan memaksimalkan nilai dari koefisien Bhattacharyya  $\hat{\rho}(y)$ . Pencarian lokasi target yang baru pada frame saat ini dimulai dari lokasi  $\hat{y}_0$  target pada frame yang sebelumnya. Oleh karena itu, probabilitas  $\{\hat{p}_u(\hat{y}_0)\}_{u=1 \dots m}$  dari target kandidat pada lokasi  $\hat{y}_0$  di frame saat ini harus dihitung terlebih dahulu. Menggunakan Taylor expansion disekitar nilai-nilai  $\hat{p}_u(\hat{y}_0)$ , pendekatan linier dari koefisien Bhattacharyya diperoleh setelah beberapa manipulasi seperti pada persamaan (2.1).

$$\rho[\hat{p}(y), \hat{q}] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_0) \hat{q}_u} + \frac{1}{2} \sum_{u=1}^m \hat{p}_u(y) \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}} \quad (2.1)$$

Pendekatan yang diinginkan adalah ketika target kandidat  $\{\hat{p}_u(\hat{y}_0)\}_{u=1 \dots m}$  tidak mengalami perubahan yang drastis dari inialisasi  $\{\hat{p}_u(\hat{y}_0)\}_{u=1 \dots m}$  yang merupakan asumsi yang paling valid antara frame secara berturut – turut Kondisi  $\hat{p}_u(\hat{y}_0) > 0$  untuk semua  $u = 1 \dots m$ , selalu dapat dipaksakan dengan tidak menggunakan nilai yang melanggar. Dengan beberapa manipulasi yang dilakukan, fungsi dari target kandidat berubah menjadi seperti pada persamaan (2.2).

$$\rho[\hat{p}(y), \hat{q}] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_0) \hat{q}_u} + \frac{c_h}{2} \sum_{i=1}^{n_h} w_i k\left(\left\|\frac{y-x_i}{h}\right\|^2\right) \quad (2.2)$$

dimana

$$w_i = \sum_{u=1}^m \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}} \delta[b(x_i) - u] \quad (2.3)$$

Dengan demikian untuk meminimalisasikan jarak, nilai dari koefisien Bhattacharyya harus dimaksimalkan. Estimasi kepadatan dihitung dengan profil kernel  $k(x)$  pada  $y$  di frame saat ini, dengan data yang dihitung oleh  $w_i$ . Pada prosedur ini seperti pada persamaan (2.2), kernel secara berulang di pindahkan dari lokasi saat ini  $\hat{y}_0$  menuju lokasi  $\hat{y}_1$  sesuai dengan persamaan (2.4).

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g\left(\left\|\frac{\hat{y}_0 - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{\hat{y}_0 - x_i}{h}\right\|^2\right)} \quad (2.4)$$

dimana  $g(x) = -k'(x)$ , dengan asumsi bahwa turunan dari  $k(x)$  ada untuk semua  $x \in [0, \infty)$ , kecuali untuk nilai terbatas.

## 2.4. Fuzzy Logic

Fuzzy merupakan sebuah keadaan yang tidak dapat ditentukan secara jelas atau pasti. Fuzzy logic system merupakan model mekanisme yang mengadopsi prinsip fuzzy[16] ini, sehingga telah banyak digunakan sebagai classifier karena kemampuannya menangani data yang bersifat ambigu, nonlinier, maupun mengandung banyak noise[16].

### 2.4.1. Type-2 Fuzzy Logic

Type-2 fuzzy logic atau sering disingkat T2 FL merupakan pengembangan dari type-1 fuzzy logic (T1FL). Type-2 fuzzy logic memiliki kemampuan yang lebih baik untuk memodelkan suatu ketidakpastian[18], sedangkan type-1 fuzzy logic hanya bisa menangani tingkat ketidakpastian yang terbatas sehingga T1 FL tidak dapat sepenuhnya menangani dan mengakomodasi ketidakpastian linguistik dan numerik yang terkait dengan perubahan dan lingkungan yang tidak terstruktur[9]. Hal tersebut merupakan dasar dari terciptanya type-2 fuzzy logic, namun type-2 fuzzy logic memiliki formasi yang lebih kompleks.

### 2.4.2. Interval Type-2 Fuzzy Logic

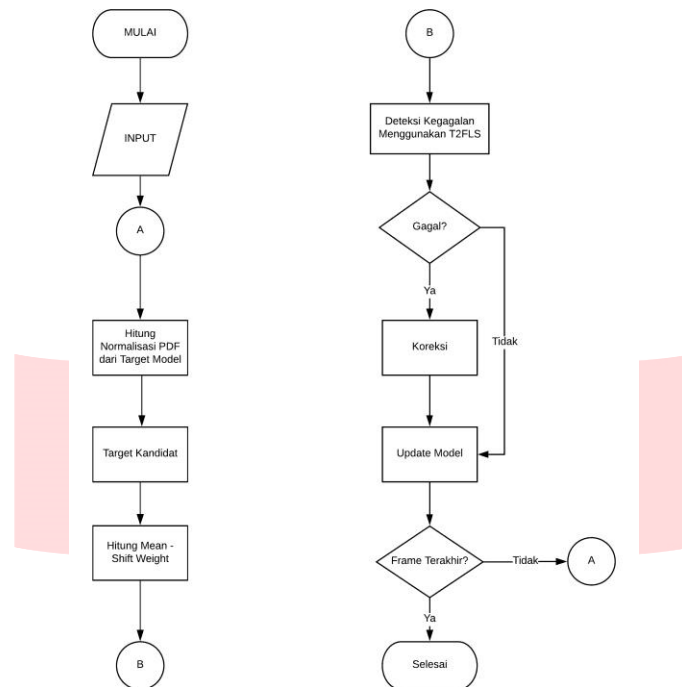
Sistem logika fuzzy pertama kali diperkenalkan oleh Prof. Lotfi Zadeh untuk memproses data[10]. T2 FL memiliki performansi yang lebih baik dari T1 FLS. Ada beberapa cara untuk memodelkan T2 FL, yaitu general T2[18], interval type-2 fuzzy logic (IT2FL)[17], dan yang terbaru Quasi Type-2 fuzzy logic[20]. General type-2 fuzzy logic sulit untuk di implementasikan karena secara sistematis masih rumit sehingga membuat komputasi menjadi kompleks. Sedangkan interval type-2 fuzzy logic sangat mudah diatur sehingga lebih mudah diimplementasikan[17].

## 3. Pembahasan

### 3.1. Desain Sistem

Dalam tugas akhir ini dirancang sebuah sistem pelacakan objek bergerak atau object tracking menggunakan video dari OTB-50 sebagai input. Metode yang digunakan adalah Kernel-based Object Tracking menggunakan Type-2 Fuzzy Logic. Type-2 Fuzzy Logic yang digunakan dalam sistem ini adalah interval type-2 fuzzy logic

dimana digunakan untuk koreksi kegagalan *tracking* pada *output* dari *Kernel-based Object Tracking*. Secara umum alur dari skema sistem kerja pada penelitian ini direpresentasikan pada gambar 3.1.



Gambar 3.1 Skema Sistem

### 3.2. Parameter Performansi

Dalam pengujian algoritma yang dirancang terdapat beberapa parameter yang dapat menentukan kualitas dari algoritma yang telah di uji, yaitu *precision plot* dan *success plot*.

#### 3.2.1. Precision Plot

*Precision plot* merupakan parameter yang digunakan untuk mengevaluasi pelacakan objek berdasarkan *center location error*. *Precision plot* menghitung rata-rata jarak *euclidean* antara lokasi tengah pada objek target pelacakan dan posisi tengah setiap *frame*. Persentase dari *frame* yang estimasi lokasinya diberikan jarak *threshold* dari *ground-truth* merupakan pengukuran performansi yang lebih baik dibandingkan tanpa adanya *threshold* jarak dari *ground-truth*. *Center location error* hanya menghitung perbedaan dan tidak memperhatikan ukuran dan skala target objek[11]. Persamaan 3.1 merupakan formula yang digunakan untuk menghitung jarak antara centroid bounding box *ground-truth* dan bounding box tracker.

$$J_c = \sqrt{q(x_t - x_g)^2 + (y_t - y_g)^2} \quad (3.1)$$

dimana  $J_c$  merupakan jarak *centroid*,  $x_t$  dan  $y_t$  merupakan titik *centroid*  $x,y$  pada *bounding box tracker*,  $x_g$  dan  $y_g$  merupakan titik *centroid*  $x,y$  pada *bounding box ground-truth*.

#### 3.2.2. Success Plot

*Success plot* merupakan parameter yang mengevaluasi pelacakan objek atau *object tracking* berdasarkan skor *overlap*. Dimisalkan *bounding box* dari objek yang dilacak  $r_t$  dan *bounding box groundtruth* direpresentasikan sebagai  $r_0$  maka skor *overlap* didenisikan sebagai persamaan 3.2 berikut[11]:

$$s = \frac{(|r_t \cap r_0|)}{(|r_t \cup r_0|)} \quad (3.2)$$

dimana  $\cap$  dan  $\cup$  merupakan irisan dan gabungan, dan  $||$  menunjukkan jumlah *pixel* dalam suatu daerah. Rata-rata dari skor *overlap* atau sering disebut dengan *average overlap score* (AOS) dapat digunakan sebagai pengukur performansi. Skor *overlap* dapat digunakan untuk menentukan apakah sebuah algoritma melacak dengan sukses atau tidak pada suatu *frame*, dengan menguji apakah  $s$  lebih besar dari *threshold* yang ditentukan  $t_0$  (misal,  $t_0 = 0.5$ )[11].

## 4. Hasil dan Analisis

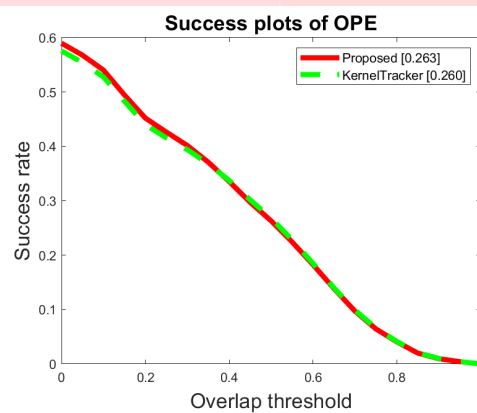
### 4.1. Inisialisasi *Bounding Box*

Untuk menjalankan sistem ini diperlukan inisialisasi objek pada *frame* pertama dari video yang akan dilacak atau sering disebut inisialisasi *ground truth*. Inisialisasi objek pada *frame* pertama dilakukan dengan cara memberikan *bounding box* pada objek yang akan dilacak. Hal ini diperlukan sebagai nilai awal atau informasi awal untuk menentukan lokasi objek pada *frame* selanjutnya di setiap *sequence*.

#### 4.1.1. Hasil Performansi Berdasarkan *Success Plot*

Berikut hasil dari performansi yang dihasilkan berdasarkan *Success Plot* dari sistem yang dibuat dengan nama *proposed* yang memiliki performansi paling baik dengan *kernel-based object tracking* sebelum ditingkatkan performansinya menggunakan IT2FL berikut atribut-atribut yang digunakan pada *Success Plot illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view, background clutter* dan *low resolution*.

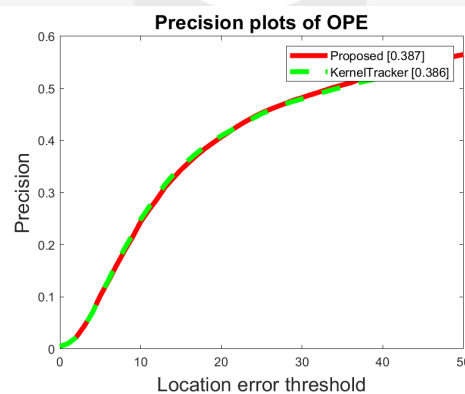
Dari seluruh *frame* yang dijalankan, yaitu sebanyak 26972 *frame*. Pada pengujian sistem yang di buat didapatkan performansi sebesar 0.263 meningkat sebesar 0.003 dari system *kernel-based object tracking*, dimana system *kernel-based object tracking* menghasilkan performansi sebesar 0.260.



Gambar 4.1 Hasil Performansi Berdasarkan *Success Plot*

#### 4.1.2. Hasil Performansi Berdasarkan *Precision Plot*

Berikut hasil dari performansi yang dihasilkan berdasarkan *Precision Plot* dari sistem yang dibuat dengan nama *proposed* yang memiliki performansi paling baik dengan *kernel-based object tracking* sebelum ditingkatkan performansinya menggunakan IT2FL berikut atribut-atribut yang digunakan pada *Success Plot illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view, background clutter* dan *low resolution*.

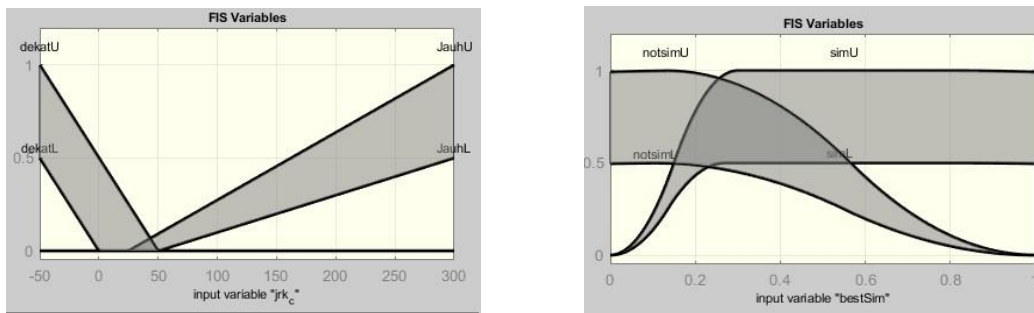


Gambar 4.2 Hasil Performansi Berdasarkan *Success Plot*

## 4.2. Analisis Arsitektur *Membership Fuzzy*

Terdapat beberapa arsitektur yang telah di uji untuk menjadi pertimbangan dalam menentukan *membership* yang memiliki performansi yang baik untuk setiap parameter masukan *type-2 fuzzy logic*. Pada bagian ini akan dijelaskan bagian tersebut.

#### 4.2.1. Membership Interval Type-2 Fuzzy logic



Gambar 4.3 Arsitektur dari *Membership Centroid* dan *Membership Similarity*

Gambar 4.3 dibagian sebelah kiri menunjukkan *membership* dari parameter *similarity* pada arsitektur fuzzy pertama dan gambar disebelah kanan menunjukkan jarak *centroid*. Pada gambar diatas *range parameter similarity* adalah dari 0 sampai 1 dan terdapat *overlap area* disekitar range 0.175 sampai 0.55, sedangkan parameter jarak *centroid* memiliki *range* dari -50 sampai 300 dan *overlap area* terjadi pada *range* sekitar 25 sampai 50. Pada pengujian performansi berdasarkan *precision plot* dan *success plot*, arsitektur fuzzy 1 dinamakan *Proposed*.

### 5. Kesimpulan dan Saran

#### 5.1. Kesimpulan

Dari beberapa hasil analisis dan pengujian yang telah dilakukan berdasarkan rumusan masalah sebelumnya, berikut kesimpulan dari penelitian ini.

1. Dalam system yang dibuat ini, *type-2 fuzzy logic* digunakan untuk mendeteksi kegagalan saat pelacakan, dimana T2FL yang digunakan ialah *interval type2 fuzzy logic* yang diaplikasikan pada metode pelacakan objek *kernel-based object tracking*.
2. Parameter yang digunakan sebagai masukan IT2FL adalah jarak *centroid* dan *similarity* antara *bounding box* saat ini dan *bounding box* sebelumnya. Dimana bentuk *membership* yang digunakan pada jarak *centroid* adalah *triangular-shaped* dengan 2 kategori yaitu dekat dan jauh. *Range* untuk kategori dekat dimulai dari -50 sampai 300 *pixel* dan kategori jauh memiliki *range* dari 25 *pixel* sampai 300 *pixel*. Sedangkan untuk *similarity* menggunakan  $\pi$ -shaped terdapat 2 kategori, yaitu mirip dan tidak mirip. Dimana kategori mirip memiliki *range* dari 0.2 sampai 1 sedangkan kategori tidak mirip memiliki *range* dari 0 sampai 0.5.
3. Dengan menggabungkan *type-2 fuzzy logic* dengan *kernel-based object tracking*, performansi dari sistem yang dibuat dapat meningkat. Berdasarkan *precision plot*, sistem yang dibuat mengalami peningkatan performansi sebesar 0.001 dan berdasarkan *success plot*, sistem yang dibuat berhasil mengalami peningkatan performansi sebesar 0.03.
4. Arsitektur *membership fuzzy* dan juga parameter yang digunakan sebagai masukan *type-2 fuzzy logic* sangat mempengaruhi performansi dari sistem yang diusulkan. Apabila arsitektur fuzzy yang digunakan salah, maka hal tersebut dapat menurunkan performansi dari *kernel-based object tracking*.

#### 5.1. Kesimpulan

Berikut beberapa saran yang dapat dijadikan referensi untuk pengembangan penelitian lebih lanjut dari tugas akhir ini.

1. Mencoba parameter lain sebagai masukan pada sistem *type-2 fuzzy logic*.
2. Bentuk *Membership* parameter fuzzy dapat diubah-ubah sehingga dapat dibandingkan performansi *kernel-based object tracking* dengan *membership* lainnya.
3. Mengaplikasi *type-2 fuzzy logic* system ke metode *object tracking* lainnya.
4. Menambah parameter sebagai masukan *type-2 fuzzy logic*
5. Pengujian sistem menggunakan data set selain OTB-50.

#### Daftar Pustaka:

- [1] Y. Zhong, A. K. Jain, and M.-P. Dubuisson-Jolly, "Object tracking using deformable templates," IEEE transactions on pattern analysis and machine intelligence, vol. 22, no. 5, pp. 544–549, 2000.
- [2] J.-Y. Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," Intel Corporation, vol. 5, no. 1-10, p. 4, 2001.
- [3] A. Yilmaz, X. Li, and M. Shah, "Contour-based object tracking with occlusion handling in video acquired using mobile cameras," IEEE Transactions on pattern analysis and machine intelligence, vol. 26, no. 11, pp. 1531–1536, 2004.

- [4] W. Wilson, L. Liliana, and K. Gunadi, "Perancangan dan pembuatan aplikasi tracking object pada video dengan metode kernel-based," *Jurnal Infra*, vol. 1, no. 1.
- [5] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 5, pp. 564–575, 2003.
- [6] K. Gunadi et al., "Perancangan dan pembuatan aplikasi tracking object pada video dengan metode kernel-based," *Jurnal Infra*, vol. 1, no. 1, pp. pp–66, 2013.
- [7] S. A. Wibowo, H. Lee, E. K. Kim, T. Kwon, and S. Kim, "Tracking failures detection and correction for face tracking by detection approach based on fuzzy coding histogram and point representation," in *2015 International Conference on Fuzzy Theory and Its Applications (iFUZZY)*. IEEE, 2015, pp. 34–39.
- [8] S. A. Wibowo, H. Lee, E. K. Kim, and S. Kim, "Collaborative learning based on convolutional features and correlation filter for visual tracking," *International Journal of Control, Automation and Systems*, vol. 16, no. 1, pp. 335–349, 2018.
- [9] O. Linda and M. Manic, "Uncertainty-robust design of interval type-2 fuzzy logic controller for delta parallel robot," *IEEE transactions on industrial informatics*, vol. 7, no. 4, pp. 661–670, 2011.
- [10] O. Castillo, P. Melin, J. Kacprzyk, and W. Pedrycz, "Type-2 fuzzy logic: theory and applications," in *2007 IEEE International Conference on Granular Computing (GRC 2007)*. IEEE, 2007, pp. 145–145.
- [11] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2411–2418.
- [12] F. A. Hermawati, "Pengolahan citra digital," ANDI. YOGYAKARTA, 2013.
- [13] R. Kusumanto and A. N. Tomponu, "Pengolahan citra digital untuk mendeteksi obyek menggunakan pengolahan warna model normalisasi rgb," *Semantik*, vol. 1, no. 1, 2011.
- [14] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "An adaptive color-based particle filter," *Image and vision computing*, vol. 21, no. 1, pp. 99–110, 2003.
- [15] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 17, no. 8, pp. 790–799, 1995.
- [16] G. J. Klir and B. Yuan, *Fuzzy sets and fuzzy logic: theory and applications*. Prentice Hall PTR New Jersey, 1995, vol. 574.
- [17] J. M. Mendel, R. I. John, and F. Liu, "Interval type-2 fuzzy logic systems made simple," *IEEE transactions on fuzzy systems*, vol. 14, no. 6, pp. 808–821, 2006.
- [18] T.-C. Lin, "Observer-based robust adaptive interval type-2 fuzzy tracking control of multivariable nonlinear systems," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 3, pp. 386–399, 2010.
- [19] J. M. Mendel, "General type-2 fuzzy logic systems made simple: a tutorial," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 5, pp. 1162–1182, 2014.
- [20] X. Ma and J. Zhan, "Generalized fuzzy h-bi-ideals and h-quasi-ideals of hemirings," *Information Sciences*, vol. 179, no. 9, pp. 1249–1268, 2009.
- [21] A. Shivhare and V. Choudhary, "Object tracking in video using mean shift algorithm: A review."