

AKURASI PENJEJAKAN OBJEK DALAM BERAGAM RUANG WARNA

Gede Sukadarmika¹⁾, Dewa Made Wiharta²⁾, Nyoman Putra Sastra³⁾

Abstract— The object trace has been a problem in estimating an object position when the object is moving due to the heavy influence of the uncertainty. Many researcher claim that color histogram is reliable feature to represent this object. . Different investigators use different color spaces in conducting research on tracking the object. So, there is no numerical comparison of the impact of the use of different color spaces to the successful tracking. This study compare the performance of tracking an object by using a different color space i.e.: RGB, HSV, and CIELAB. The performance is shown numerically by comparing the actual position of the object with the results of the estimation.

Intisari— Penjejakan obyek adalah permasalahan dalam estimasi posisi suatu obyek saat obyek tersebut bergerak, yang banyak dipengaruhi oleh ketidak-pastian. Banyak peneliti yang menyatakan bahwa fitur yang handal digunakan untuk mewakili obyek tersebut adalah histogram warna. Peneliti berbeda menggunakan ruang warna yang berbeda dalam melakukan penelitian mengenai penjejakan obyek. Jadi, tidak ada perbandingan secara numerik mengenai dampak pemakaian ruang warna yang berbeda terhadap keberhasilan penjejakan. Dalam penelitian ini, akan dibandingkan unjuk kerja penjejakan obyek dengan menggunakan ruang warna berbeda, yaitu RGB, HSV, dan CIELAB. Unjuk kerja ditampilkan secara numerik dengan membandingkan posisi obyek sebenarnya dengan hasil estimasi

Kata Kunci — Penjejakan objek, ruang warna, RGB, HSV, CIELAB

I. PENDAHULUAN

Penjejakan obyek bisa didefinisikan sebagai suatu permasalahan terhadap estimasi pergerakan suatu obyek dalam bidang citra. Penjejakan obyek bisa menjadi suatu algoritma yang kompleks yang diakibatkan oleh noise dalam citra, hilangnya informasi akibat proyeksi dari dunia 3D pada citra 2D, pergerakan obyek yang rumit, halangan (sebagian atau keseluruhan) dsb. Beragamnya kendala tersebut memaksa para peneliti untuk memberikan batasan pada algoritma yang mereka kembangkan.

^{1,2,3} Dosen Jurusan Teknik Elektro dan Komputer Fakultas Teknik Universitas Udayana, Jln. Jalan Kampus Bukit Jimbaran 80361 INDONESIA (telp: 0361-703315; fax: 0361-4321; Email [sukadarmika, wiharta,putra.sastra]@unud.ac.id

Dari sejumlah pendekatan yang pernah dilakukan, metode dengan menggunakan deskriptor berbasis histogram warna telah banyak mendapat perhatian. (Laptev, 2009; Lu dkk., 2009). Algoritma penjejakan berbasis histogram warna telah diterapkan dalam 2 (dua) pendekatan yang berbeda. Pertama, teknik mean shift (Comaniciu dkk., 2003) berusaha untuk meminimalkan jarak histogram warna yang diberi bobot dengan minimisasi berbasis gradien. Pencarian yang dilakukan adalah bersifat deterministik. Metode yang kedua dilakukan dalam kerangka filter partikel (Laptev, 2009; Nummiaro dkk., 2003; Wiharta dkk., 2011). Filter partikel melakukan penjejakan dengan cara melemparkan sejumlah partikel yang diberi bobot sesuai dengan suatu fungsi *likelihood* yang telah ditentukan. Posisi obyek yang dicari didapatkan dari rata-rata nilai partikel. Jadi, probabilitas kerapatan posterior dari lokasi obyek didiskritkan dalam sejumlah partikel yang diberi bobot. Dalam hal ini, optimisasi mean shift diganti dengan evolusi sejumlah partikel yang bersifat probabilistik dengan rata-rata bobotnya.

Ada sejumlah ruang warna yang bisa digunakan dalam membangun histogram warna, seperti RGB, HSV, atau CIELAB. Masing-masing ruang warna tersebut mempunyai karakteristik yang berbeda. Untuk bisa melihat unjuk kerja beragam ruang warna dalam konteks penjejakan obyek, maka perlu dikembangkan suatu aplikasi penjejakan obyek dengan menggunakan fitur histogram dari beberapa ruang warna yang dipilih. Paper ini membandingkan pemakaian beberapa ruang warna dalam membangun histogram yang dipergunakan sebagai fitur dalam penjejakan obyek.

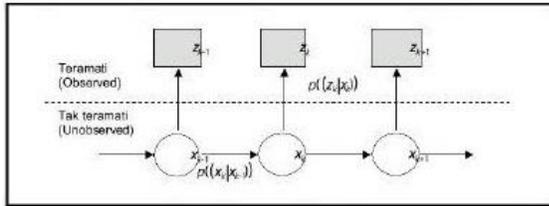
II. TINJAUAN PUSTAKA

Suatu sistem stokastik nonlinear bisa didefinisikan dengan persamaan dinamis (model proses/evolusi) ruang waktu diskrit :

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{w}_{k-1}) \quad (1)$$

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{v}_k) \quad (2)$$

dimana $f(\cdot)$ dan $h(\cdot)$ adalah fungsi vektor, \mathbf{x}_k adalah vektor *state* sistem, \mathbf{w}_k adalah vektor noise dinamis, \mathbf{z}_k adalah vektor pengukuran, dan \mathbf{v}_k adalah vektor noise pengukuran. Model probabilistik ekuivalen dari sistem dinamik (1) dan (2) ditunjukkan dalam model grafik pada gambar 1.



Gambar 1. Model Probabilistik ekuivalen dari system dinamik 1 dan 2

Kerapatan $p(\mathbf{z}_k|\mathbf{x}_k)$ menyatakan probabilitas pengamatan \mathbf{z}_k , ketika sistem berada dalam *state* \mathbf{x}_k . Kerapatan $p(\mathbf{z}_k|\mathbf{x}_k)$ juga merupakan *likelihood* sistem berada dalam *state* \mathbf{x}_k , dengan diberikan pengamatan \mathbf{z}_k . Dalam teori filtering, $p(\mathbf{z}_k|\mathbf{x}_k)$ seringkali disebut dengan nama fungsi *likelihood*. Kerapatan transisi $p(\mathbf{x}_k|\mathbf{x}_{1:k})$ didefinisikan secara lengkap dengan fungsi transisi $f(\mathbf{x}_k, \mathbf{w}_k)$ dan distribusi noise proses $p(\mathbf{w}_k)$, sedangkan fungsi *likelihood* $p(\mathbf{z}_k|\mathbf{x}_k)$ dispesifikasikan dengan fungsi pengamatan $h(\mathbf{x}_k, \mathbf{v}_k)$ dan distribusi noise pengamatan $p(\mathbf{v}_k)$.

Dari perspektif Bayesian, permasalahan penjejukan adalah untuk menghitung derajat keyakinan (*degree of belief*) secara rekursif dalam *state* \mathbf{x}_k pada waktu ke- k , dengan diberikan data $\mathbf{z}_{1:k}$. Dalam literatur (Candy, 2009; Arulampalam dkk. 2002), rekursi dari filter bayes dibagi menjadi dua tahap, yaitu : tahap prediksi dan update. Dalam tahap prediksi, posterior $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ dari waktu sebelumnya ($k-1$) dipropagasi melalui model dinamis $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ untuk menghasilkan distribusi prediktif $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ dengan menggunakan relasi Chapman-Kolmogorov (Arulampalam dkk. 2002) :

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int_{-\infty}^{\infty} p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_{1:k-1}) p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1} \quad (3)$$

Persamaan (1) menyatakan suatu proses Markov orde 1, maka $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_{1:k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1})$ dan persamaan (3) dapat disederhanakan menjadi :

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int_{-\infty}^{\infty} p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1} \quad (4)$$

Dalam tahap *update*, ketika data pengamatan baru tersedia, distribusi prediktif diperbaharui dengan menggunakan *likelihood* $p(\mathbf{z}_k|\mathbf{x}_k)$ dan dinormalisasi untuk menghasilkan posterior yang baru $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ dengan menggunakan aturan Bayes :

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k | \mathbf{x}_k)p(\mathbf{x}_k | \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k | \mathbf{z}_{1:k-1})} \quad (5)$$

dimana konstanta normalisasi dihitung dengan integrasi penyebut pada semua nilai \mathbf{x}_k :

$$p(\mathbf{z}_k | \mathbf{z}_{1:k-1}) = \int_{-\infty}^{\infty} p(\mathbf{z}_k | \mathbf{x}_k)p(\mathbf{x}_k | \mathbf{z}_{1:k-1})d\mathbf{x}_k \quad (6)$$

yang tergantung dari fungsi *likelihood* $p(\mathbf{z}_k|\mathbf{x}_k)$ yang didefinisikan oleh model pengukuran (2) dan statistik dari \mathbf{v}_k yang diketahui.

Hubungan antara (4) dan (5) membentuk dasar dari solusi Bayesian yang optimal. Propagasi rekursif dari kerapatan posterior ini hanyalah solusi konseptual karena secara umum tidak bisa diselesaikan secara analitik (Arulampalam dkk., 2002). Solusi tersedia dalam kasus-kasus terbatas, antara lain filter Kalman dan filter grid-based.

Kalau deretan noise adalah Gaussian dan f dan h adalah fungsi linear, solusi optimal diberikan oleh filter Kalman (Kalman, 1960), yang menghasilkan posterior yang juga Gaussian. Dalam filter Kalman, rekursi persamaan (4) dan (5) bisa diinterpretasikan sebagai estimasi rekursif dari mean dan kovarians dari posterior $p(\mathbf{x}_k|\mathbf{z}_{1:k})$.

Dengan berkembangnya kemampuan komputasi komputer, langkah-langkah yang signifikan telah dilakukan untuk perhitungan rekursif dari filter Bayes dengan menggunakan simulasi (Gordon dkk., 1993). Sebagai hasilnya, sejumlah metode rekursi Bayesian berbasis simulasi sekuensial Monte Carlo telah dikembangkan dalam berbagai disiplin ilmu seperti statistik, ekonomi, rekayasa teknik, dan ilmu komputer. Metode sekuensial Monte carlo ini dikenal dengan dengan nama *filter partikel*.

2.1 Filter Partikel

Algoritma Sequential Important Sampling (SIS) adalah suatu metode Monte Carlo (MC) yang menjadi dasar dari filter MC yang telah dikembangkan dalam dasawarsa terakhir (Doucet dkk. 2000; Doucet dkk. 2001). Pendekatan sekuensial MC ini dikenal dengan berbagai nama, antara lain filter bootstrap (Gordon dkk. 1993), algoritma kondensasi (MacCormick and Blake. 2000), filter partikel (Carpenter dkk. 1999), interacting particle approximation, dan survival of the fittest (Kanazawa dkk. 1995). Pendekatan ini merupakan suatu teknik untuk implementasi sebuah filter Bayesian rekursif dengan simulasi Monte Carlo. Ide dasar dari pendekatan ini adalah dengan menyajikan fungsi kerapatan posterior yang dibutuhkan dengan suatu kumpulan sample acak yang diberikan bobot dan menghitung estimasi berdasarkan pada sample dan bobot tersebut. Dengan meningkatkan jumlah sampel, karakteristik Monte Carlo ini menjadi representasi ekivalen dari deskripsi fungsi posterior pdf, dan filter SIS mendekati estimasi Bayesian yang optimal.

Misal $\{\mathbf{x}_{0:k}^i, \mathbf{w}_k^i\}_{i=1}^N$ menyatakan ukuran acak (*random measure*) yang memberikan karakteristik pdf posterior $p(\mathbf{x}_{0:k} | \mathbf{z}_{1:k})$, dimana $\{\mathbf{x}_{0:k}^i, i = 0, \dots, N\}$ adalah sekumpulan *support points* atau partikel yang diberikan bobot $\{\mathbf{w}_k^i, i = 0, \dots, N\}$ dan $\mathbf{x}_{0:k} = \{\mathbf{x}_j, j = 0, \dots, k\}$ adalah himpunan dari semua kondisi (*state*) sampai pada waktu ke- k . Bobot dinormalisasi sehingga $\sum_i \mathbf{w}_k^i = 1$. Dengan demikian, bisa diberikan pendekatan pada kerapatan posterior pada waktu ke- k sbb:

$$p(\mathbf{x}_{0:k} | \mathbf{z}_{1:k}) \approx \sum_{i=1}^N \mathbf{w}_k^i \delta(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^i) \quad (7)$$

Dengan demikian bisa diperoleh pendekatan diskrit terhadap posterior sebenarnya, $p(\mathbf{x}_{0:k} | \mathbf{z}_{1:k})$. Bobot-bobot dipilih menggunakan prinsip *importance sampling* (Doucet dkk., 2000).

2.1.1 Penjejakan Obyek dengan Filter Partikel

Dalam penelitian ini, akan diterapkan kerangka kerja filter partikel untuk penjejakan obyek dalam video. Tahap pertama dilakukan dalam penjejakan obyek ini adalah mendefinisikan obyek yang hendak dijejaki (inisialisasi). Inisialisasi ini dilakukan secara manual, dengan memberi bingkai persegi pada obyek yang bersangkutan pada frame pertama dari video, dengan asumsi bahwa obyek telah hadir pada frame tersebut. Metode seperti ini mempunyai kelebihan dimana pengguna mempunyai kebebasan dalam menentukan obyek yang akan dijejaki.

Bingkai persegi tersebut akan memberikan informasi prior yang menjadi pedoman bagi filter partikel dalam menentukan

lokasi partikel yang akan disebar. Jumlah partikel yang disebar akan berpengaruh pada beban komputasi. Pada penelitian awal yang sudah dilakukan, partikel sejumlah 200-300 sudah mencukupi untuk bisa memberikan deskripsi mengenai obyek terkait, dengan persyaratan *real-time* masih terpenuhi.

Bobot dari masing-masing partikel akan tergantung dari fungsi likelihood yang diberikan. Jika fungsi likelihood $f(y|x)$ dan prior $p(x)$ diketahui, adalah mudah untuk menghitung distribusi posterior dari $p(x|y)$, yang dipergunakan untuk mendapatkan *inferensi*, dalam hal ini posisi obyek dalam frame berikutnya. Pada kondisi awal, dimana informasi mengenai bobot partikel yang harus diberikan belum diketahui, digunakan prinsip *noninformative prior*. Suatu pertanyaan yang penting dan mendasar adalah bagaimana cara untuk mendapatkan *noninformative prior*. Bayes/Laplace telah menyatakan bahwa jika tidak ada yang diketahui mengenai x , maka prior $p(x)$ dinyatakan mempunyai distribusi uniform, yaitu seluruh *outcome* dari x yang mungkin mempunyai probabilitas yang sama. Ini juga dikenal dengan nama "*principle of insufficient reason*" (Datta dan Ghosh, 1996). Jadi, berdasarkan teori di atas, pembobotan awal partikel dilakukan dengan menggunakan distribusi uniform,

$$w = 1/N$$

dimana N adalah jumlah partikel yang digunakan.

Tahap berikutnya adalah propagasi partikel sesuai dengan model pergerakan yang dipergunakan. Bobot partikel diperbaharui (*update*) dengan menggunakan fungsi likelihood, dimana fungsi likelihood ini akan tergantung dari fitur obyek yang digunakan. Posisi obyek pada frame berikutnya bisa diperoleh dengan menggunakan nilai rata-rata bobot partikel. Keandalan dari algoritma penjejakan obyek ini banyak tergantung dari fitur obyek yang digunakan. Dalam sub-bab berikut, akan dijabarkan mengenai fitur obyek yang menjadi dasar dari algoritma penjejakan obyek yang akan dikembangkan.

2.1.2 Penjejakan obyek Berbasis Histogram Warna

Bagian penting dari penjejakan visual adalah model visual dari obyek. Untuk obyek yang nonrigid (tidak kaku, bisa berubah bentuk) seperti manusia, visual model menimbulkan permasalahan karena obyek seringkali berubah bentuk. Pendekatan yang bisa digunakan adalah menggunakan model visual yang tidak detail, yaitu histogram warna (Nummiaro, 2003). Dalam sub-bab ini akan dibahas mengenai pemodelan obyek yang berbasis histogram warna.

2.2 Representasi Obyek

Suatu ruang fitur dipilih pertama kali untuk memberikan karakteristik mengenai obyek target. Model target referensi disajikan dengan pdf q dalam ruang fitur. Sebagai contoh, model referensi yang dipilih bisa berupa pdf warna dari target.

Model target didefinisikan berada pada lokasi spasial 0. Dalam frame berikutnya, sebuah kandidat target didefinisikan berada pada lokasi y , dan dikarakteristikkan dengan pdf $p(y)$ (Nummiaro dkk., 2003; Comanicu dkk.,2003). Histogram bukan merupakan estimasi kerapatan non-parametrik terbaik, tapi mencukupi untuk menangani permasalahan yang dihadapi.

2.3 Model Target dan Kandidat Target

Untuk mengurangi beban komputasi yang diakibatkan oleh pengolahan real-time terhadap kerapatan diskrit, digunakan sejumlah m bins histogram. Dengan demikian diperoleh model target (Comanicu dkk.,2003):

$$\hat{q} = \{\hat{q}_u\}_{u=1\dots m} \tag{8}$$

$$\sum_{u=1}^m \hat{q}_u = 1 \tag{9}$$

Target disajikan dengan suatu daerah persegi (bisa juga dengan daerah elipse) dalam ruang citra. Diberikan $\{\mathbf{x}_i^*\}_{i=1\dots n}$ yang merupakan lokasi piksel ternormalisasi dalam daerah persegi yang didefinisikan sebagai model target. Daerah persegi ini berpusat di 0,0. Suatu kernel isotropic, dengan profil kernel $k(x)$ yang selalu berkurang (*monotonic decreasing*) menangani bobot yang lebih kecil pada piksel yang lebih jauh dari pusat obyek. Profil dari kernel K didefinisikan sebagai fungsi $k \mid 0 \leq k < \infty \rightarrow \mathbb{R}$ sedemikian hingga $K(x) = k(\|x\|^2)$. Penggunaan pembobotan semacam ini akan meningkatkan kehandalan pada estimasi kerapatan karena piksel yang berada jauh dari pusat obyek seringkali dipengaruhi oleh penghalang atau terkena interferensi dari background.

Fungsi $b : \mathbb{R}^2 \rightarrow \{1 \dots m\}$ memberikan asosiasi indeks $b\{\mathbf{x}_i^*\}$ pada piksel pada lokasi \mathbf{x}_i^* dalam ruang fitur yang terkuantisasi. Probabilitas dari fitur $u = 1, \dots, m$ dalam model target kemudian dihitung dengan (Comanicu dkk.,2003):

$$\hat{q}_u = C \sum_{i=1}^n (k \parallel \mathbf{x}_i^* \parallel^2) \delta[b(\mathbf{x}_i^*) - u] \tag{10}$$

dimana δ adalah fungsi delta Kronecker. Konstanta normalisasi C diturunkan dengan memberikan persyaratan $\sum_{u=1}^m \hat{q}_u = 1$. Karena penjumlahan fungsi delta untuk $u = 1, \dots, m$ adalah 1 (satu) maka (Comanicu dkk.,2003) :

$$C = \frac{1}{\sum_{i=1}^n k(\parallel \mathbf{x}_i^* \parallel^2)} \tag{11}$$

2.4 Konsep Ruang Warna

Suatu ruang warna merupakan representasi matematik dari suatu set warna. Tipikal warna dari suatu gambar dinyatakan sebagai layer Merah (R), Hijau (G) dan Biru (B) atau sering

disebut Ruang Warna RGB. Ruang pewarnaan RGB paling banyak digunakan dalam komputer grafik karena layar warna menggunakan warna dasar RGB untuk membentuk warna yang diinginkan [13]. Dengan demikian penggunaan space warna RGB akan menyederhanakan desain system dan arsitekturnya. Ruang warna RGB juga memiliki informasi luminance dan ketajaman (brightness) pada setiap layer dari ruang warna RGB. Variasi level brightnes dari suatu citra menyebabkan perubahan nilai RGB-nya sehingga menyebabkan ketidakstabilan dari citra tersebut [14]. Sebagai upaya untuk menghilangkan efek brightness, ruang warna RGB ditransformasikan menjadi ruang warna yang lain yang akan memisahkan efek brightness dengan informasi warna. Tipikal ruang warna yang lain yang digunakan pada area video tracking dan video surveillance adalah CIE [13] dan HSV [15].

HSV Color Space

$$\begin{aligned} L^* &= 116F(Y/Y_n) - 16 \\ a^* &= 500[f(x/X_n) - f(Y/Y_n)] \\ b^* &= 200[f(Y/Y_n) - f(Z/Z_n)] \end{aligned}$$

Ruang warna HSV memiliki property space warna yang telah memisahkan informasi brightness dengan informasi warnanya. Pada ruang warna HSV, informasi luminannya ditempatkan pada layer V, sedangkan informasi chromaticnya di tempatkan pada layer H (hue) dan S (saturation). Hubungan antara RGB dengan HSV diberikan pada persamaan berikut [15]

$$\begin{aligned} S &= (\max(R,G,B) - \min(R,G,B))/\max(R,G,B) \\ V &= \max(R,G,B) \end{aligned}$$

H = tergantung pada yang mana dari R, G, B bernilai maximum

$$\begin{aligned} \text{Ditentukan : } X &= \min(R,G,B) \\ r &= (V-R)/(V-X) \\ g &= (V-G)/(V-X) \\ b &= (V-B)/(V-X) \end{aligned}$$

Kemudian

$$\begin{aligned} H &= 5 + b \text{ if } (R=V \text{ and } G = X) \\ H &= 1 - g \text{ if } (R=V \text{ and } G \neq X) \\ H &= 1 + r \text{ if } (G = V \text{ and } B = X) \\ H &= 3 - b \text{ if } (G = V \text{ and } B \neq X) \\ H &= 3 + g \text{ if } (R = X) \\ H &= 5 - r \text{ if } (R \neq X) \end{aligned}$$

Hue (H) selanjutnya dikonversikan ke dalam derajat dengan cara mengalikannya dengan 60.

CIELab

CIE 1976 $L^*a^*b^*$ atau ruang warna CIELab menggunakan lightness (L), red-greenness (a) dan yellow-blueness (b) sebagai tiga nilai stimulusnya. Ini membuat penyesuaian dengan ruang

warna CIE XYZ untuk membuat persepsi linear, sehingga pada saat pergerakan melewati grafik, perubahan warna akan terjadi pada laju yang konstan. Hubungan tidak linear untuk L^* , a^* , dan b^* diharapkan meniru respon dari mata. CIELab secara langsung didasari oleh CIE XYZ [16] :

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{aligned} L^* &= 116F(Y/Y_n) - 16 \\ a^* &= 500[f(x/X_n) - f(Y/Y_n)] \\ b^* &= 200[f(Y/Y_n) - f(Z/Z_n)] \end{aligned}$$

Dimana

$$f(t) = \begin{cases} t^{1/3} & \text{jika } t > \left(\frac{6}{29}\right)^3 \\ \frac{1}{3} \left(\frac{29}{6}\right)^2 t + \frac{4}{29} & \text{untuk nilai } t \text{ lainnya} \end{cases}$$

Di sini X_n , Y_n and Z_n adalah tiga nilai stimulus dari referensi putih.

III. METODE

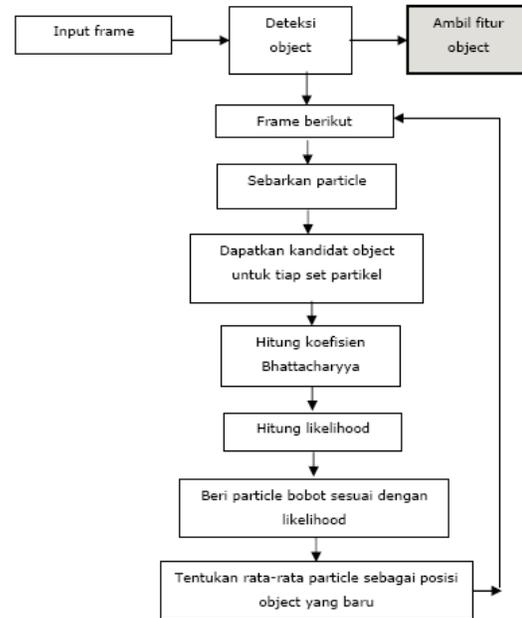
Sistem penjejakan obyek dibangun dengan menggunakan bahasa pemrograman Matlab. Obyek yang dipantau (dijejaki) adalah obyek berwarna. Obyek yang hendak dipantau diinisialisasi secara manual dalam suatu bingkai persegi atau elips. Fitur yang digunakan adalah fitur histogram warna, dimana ruang warna yang dipilih adalah ruang warna RGB, HSV dan CIELAB. Model pergerakan menggunakan model dinamis auto regresi orde kedua. Penjejakan dilakukan dalam kerangka filter partikel dalam video berwarna. Flow chart penelitian bisa dilihat pada gambar 2.

Fokus dari penelitian ini ada pada bagian blok yang diarsir, dimana fitur obyek yang dipergunakan adalah histogram warna dari obyek. Algoritma komputasi yang dipergunakan ditunjukkan pada Algoritma 1.

Algoritma 1. Penjejakan Obyek dengan Filter Partikel :

1. Tentukan obyek yang hendak dijejaki
2. Tentukan ruang warna yang digunakan
3. Hitung distribusi target model (q)
4. Ambil frame pertama video

5. Propagasi partikel dari kumpulan S_{k-1}
6. Hitung kandidat $p_{ii}(y)$ untuk tiap sampel set
7. Hitung koefisien Bhattacharyya
8. Hitung *likelihood* koefisien Bhattacharyya
9. Beri bobot masing-masing partikel sesuai dengan *likelihood*
10. Resampling
11. Hitung lokasi baru sebagai rata-rata dari partikel yang diboboti
12. Frame berikutnya (langkah 4)



Gambar 2 . Flow chart penelitian

IV. HASIL DAN DISKUSI

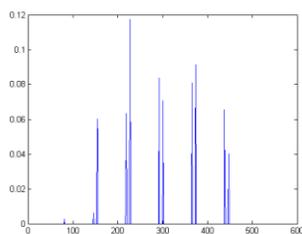
Pengujian penjejakan obyek dengan metode filter partikel telah dilakukan dengan menggunakan video uji permainan sepak bola. Sebelum pengujian dilakukan, lintasan obyek dicari secara manual untuk digunakan sebagai pembanding dalam perhitungan unjuk kerja metode penjejakan obyek. Video uji mempunyai durasi 424 frame, dan pengujian dilakukan mulai



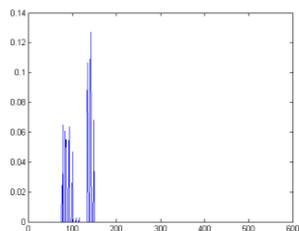
Gambar 3. Pemain sepakbola sebagai obyek yang dijejaki

dari frame 120, dimana pemain yang dijadikan obyek penjejakan mulai muncul. Penjejakan obyek ini dilakukan dalam rentang 200 frame.

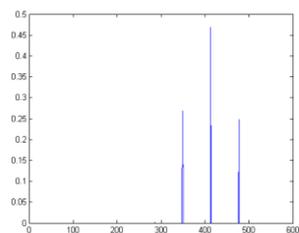
Obyek yang dipantau adalah pemain sepakbola dengan seragam putih, ditunjukkan dalam Gambar 3. Fitur yang dipergunakan untuk mewakili obyek tersebut adalah histogram warna. Gambar 4.2 menunjukkan histogram dari obyek dalam beberapa ruang warna.



(a) RGB



(b) HSV



(c) LAB

Gambar 4. Histogram obyek

Unjuk kerja penjejakan obyek dengan metode filter partikel ini dinyatakan dengan error lintasan yang diperoleh dengan menggunakan persamaan berikut :

$$\text{error} = \sqrt{\frac{\sum (\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2}{n}}$$

dimana \hat{x}_i, \hat{y}_i adalah posisi obyek dari frame ke- i , dan x_i, y_i adalah posisi obyek sebenarnya.

Setiap kali pengujian dilakukan akan memberikan hasil yang sedikit berbeda, karena sifat dari metode filter partikel yang menyebarkan partikel secara acak. Untuk itu, pengujian dilakukan sebanyak 25 kali dan hasilnya dirata-rata. Untuk ruang warna RGB dan HSV digunakan konfigurasi bin $8 \times 8 \times 8$, sedangkan konfigurasi untuk ruang warna CIELAB digunakan bin $4 \times 16 \times 8$. Jumlah partikel yang digunakan adalah 100. Hasil pengujian penjejakan obyek dengan menggunakan metode filter partikel dalam ruang warna RGB, HSV, dan CIELAB ditunjukkan pada tabel di bawah ini.

Tabel 1. Hasil Pengujian

No	Ruang Warna	Kesalahan Penjejakan
1	RGB	10.9369
2	HSV	8.914
3	CIELAB	6.5067

Dari pengujian yang dilakukan, diperoleh bahwa ruang warna CIELAB menghasilkan error terkecil dibandingkan dengan dua ruang warna lainnya, seperti ditunjukkan pada Tabel 1. Berbeda dengan ruang warna RGB, ruang warna LAB didisain untuk bisa mendekati visualisasi mata manusia. Komponen L dari CIELAB sangat mendekati persepsi mata manusia terhadap cahaya dan bisa dipergunakan untuk menghasilkan koreksi keseimbangan warna.

V. KESIMPULAN

Dalam konteks penjejakan obyek, fitur yang handal digunakan untuk mewakili obyek adalah histogram warna. Dari beragam ruang warna yang ada, seperti RGB, HSV, dan CIELAB, ruang warna CIELAB memberikan hasil penjejakan yang paling baik dilihat dari kesalahan lintasan penjejakan. Hasil ini diperoleh dari pengujian video sepak bola seperti dijabarkan pada sub bab sebelumnya. Hasil ini belum bisa secara umum menyatakan bahwa ruang warna CIELAB lebih baik dibandingkan dengan ruang warna lainnya. Pengujian lebih lanjut terhadap video lain dengan karakteristik yang berbeda diperlukan untuk bisa memberikan kesimpulan yang lebih baik.

DAFTAR PUSTAKA

1. Arulampalam, M. S., Maskell, S., Gordon, N., Clapp, T. (2002), "A Tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking" IEEE Transactions on Signal Processing 50.
2. Candy J. V. (2009). "Bayesian Signal Processing : Classical, Modern, and Particle Filtering Methods", JohnWiley & Sons, Inc., New Jersey.
3. Comaniciu, D., Ramesh, V., dan Meer, P. (2003), "Kernel-based object tracking", IEEE Transactions on Pattern Analysis and Machine Intelligence 25: 564-577.

4. Doucet, A., de Freitas, N., dan Gordon, N. (2001), "An introduction to sequential Monte Carlo methods", in *Sequential Monte Carlo Methods in Practice*, New York: Springer-Verlag.
5. Doucet, A., Godsill, S., dan Andrieu, C. (2000), "On Sequential Monte Carlo sampling methods for Bayesian filtering", *Statistics and Computing* 10(3): 197-208.
6. Gordon, N.J., Salmond, D.J., dan Smith, A.F.M. (1993), "Novel approach to nonlinear and non-Gaussian state estimation", *Proc.Inst. Elect. Eng.*
7. Kailath, T. (1967), "The divergence and Bhattacharyya distance measures in signal selection", *IEEE Trans.Commun. Tech.* 15: 52-60.
8. Lu, W., Okuma, K., dan Little, J. (2009), "Tracking and recognizing actions of multiple hockey players using the boosted particle filter", *Image and Vision Computing* 27: 16.
9. Nummiaro, K., Koller-Meier, E., dan Van Gool L. (2003), "An adaptive color-based particle filter", *Image Vision and Computing*: 99-110.
10. Wiharta, D.M., Hendrantoro, G., dan Wirawan (2010), "NonlinearNon-Gaussian State Estimation Using Particle Filter", *Proceedings The 1st International Conference on Sustainable Technology Development (ICSTD)*, Bali.
11. Yilmaz, A., Javed O., Shah M. (2006), "Object Tracking: A Survey", *ACM Computing Surveys*, Vol. 38, No. 4.
12. Laptev, I. (2009). Improving object detection with boosted histograms. *Image Vis. Comput.* 27, 535–544.