

低消費エネルギー物体追跡システムにおける 動的フレームレート制御法の改良

井上 優良^{1,a)} 小野 貴継^{2,b)} 井上 弘士^{2,c)}

概要：オンライン物体追跡は画像処理の分野において必要不可欠な技術である。物体追跡システムは、運転支援システムやスマートフォン、ドローンといったバッテリー駆動のデバイスに適用される傾向にあるため消費エネルギーの削減が求められている。この問題を解決するために、これまでに我々は動的なフレームレート制御手法を提案している。この手法は、フレーム読出しと物体追跡の2つの処理のエネルギートレードオフを考慮した上で、追跡対象物体の移動速度に応じてフレームレートを最適化する。追跡精度改善のために、この手法は実行前に与える静的なパラメータに応じてフレームレートを選択する。しかしながら、入力動画中の対象物体の種類やその動きに応じて必要となるパラメータ値は異なる。そこで本稿では、撮影状況に応じてこのパラメータを動的に調節する手法を提案する。シミュレーションによる評価の結果、大幅な追跡精度の悪化無しに、フレームレート固定の一般的な物体追跡システムに対して最大で64.3%、平均で45.0%の消費エネルギー削減を達成した。また、これまでに提案してきた手法と比較すると、平均で13.2%の消費エネルギー削減を見込めることが分かった。

キーワード：物体追跡システム、低消費エネルギー化、フレームレート

1. はじめに

動画を撮影しながら動画中の物体を追跡するオンライン物体追跡の技術は、自動運転支援システムや、拡張現実といった最新のアプリケーションに応用されている。この技術は自動車の制御機器やスマートフォンといったバッテリー駆動のシステムに組み込まれる傾向にあるため、追跡精度の向上のみならず、システムの消費エネルギー削減は重要な課題の1つである。

オンライン物体追跡は主に「フレームの読出し」と「物体追跡」という2つの処理でエネルギーを消費しており、この2つの処理量を減らすことが求められる。消費エネルギーの削減を目的として、これまで低電力のイメージセンサー [1], [2], [3], [4], [5] や処理プロセッサ [6], [7], [8], [9] が提案されてきた。しかしながら、上記の2つの処理の間に生じるエネルギートレードオフがあるため、各デバイスごとの最適化では物体追跡システムの消費エネルギー削減

効果が限定的である。たとえば、高いフレームレートで処理した場合、より多くのフレームを取得するために消費エネルギーは増加するこれに対し、フレーム間で物体の移動距離が短くなり、処理量が減ることで追跡処理に必要なエネルギーは減少する。

この問題を解決するために、我々はこれまでに動的フレームレート制御手法 [10], [11] を提案してきた。この手法は、フレームレートを消費エネルギーの制御ノブとして扱う。これにより、処理するフレーム量を調節することで、デバイス間のエネルギートレードオフを考慮した制御が可能となる。実際には、対象物体の移動速度に応じてフレームレートを決定する。たとえば、物体の移動が速い場合は、高い追跡精度を維持するために高フレームレート、つまり、より多くのフレーム読出しを処理する。反対に、物体が全く動かないなどの速度が低い場合は、フレームレートを落とすことでフレーム読出し量を減らすことで消費エネルギー削減に貢献する。高い追跡精度を達成しつつ消費エネルギーを削減するためには、適切なフレームレートの制御が必要不可欠である。フレームレートを低下させることは、処理フレームの時間間隔を広げることになる。物体の移動速度が高いにもかかわらずフレームレートを誤って低く設定した場合、次のフレームを処理する間に物体が予測よりも大きく移動することで推定を失敗する可能性が高

¹ 九州大学 大学院システム情報科学府 〒 819-0395 福岡県福岡市西区元岡 744

² 九州大学 大学院システム情報科学府 〒 819-0395 福岡県福岡市西区元岡 744

a) yusuke.inoue@cpc.ait.kyushu-u.ac.jp

b) takatsugu.ono@cpc.ait.kyushu-u.ac.jp

c) inoue@ait.kyushu-u.ac.jp

まる。このような影響を軽減するために、フレームレートを決定する際には速度が安定して推移しているか否かを判断する。この判定に基づいて、速度変化が著しい場合はフレームレートの低下を抑制し、追跡精度を維持することを試みる。

この保守的な制御方法では、実行前に与えられる2つのパラメータに基づいてフレームレートを決定している。しかしながら、実際には撮影している対象やその動きといった撮影状況はそれぞれで異なるため、さらなる消費エネルギー削減の機会を逃していると考えられる。そこで本稿では、入力動画の撮影状況に基づいてこれらのパラメータを動的に調節する方法を提案する。提案手法は撮影状況の一つである物体の動く速さに基づいて、フレームレートを変更する度に制御パラメータを動的に設定することを試みる。この提案手法を適用した結果、追跡精度の大幅な悪化無しにフレームレート固定の従来手法よりも最大で64.3%、平均で45.0%の消費エネルギーを削減を達成した。

本稿の構成は次の通りである。まず、第2節ではこれまでに我々が提案してきた動的フレームレート制御手法について説明する。次に第3節では、撮影状況に基づいた消費エネルギー削減率改善手法について提案する。続いて、第4節では提案した手法について追跡精度と消費エネルギーの観点で定量的評価を行い、最後に第5節でまとめる。

2. 動的フレームレート制御

2.1 コンセプト

本節では、我々が提案している動的フレームレート最適化制御 [10], [11] の概要について述べる。従来のオンライン物体追跡システムはマッチング処理の枝刈りや探索範囲の制限といった空間方向の計算コストの削減手法が取られてきた。これに対して、動的フレームレート最適化手法は従来の実装と異なり、追跡している物体の動きに基づいたエネルギートレードオフを考慮してフレームレートを動的に制御する。この手法で重要なことは、追跡失敗するほどの精度低下を引き起こすことなく消費エネルギーを最小化するフレームレートをいかにして決定するか、という点である。

まず一般的な物体追跡システムの処理フローについて述べる。物体追跡は「フレーム取得」と「追跡処理」という二つの処理を繰り返すことで実現される。まず「フレーム取得」では、物体追跡システムはメモリから1フレーム読み出す処理を行う。この際、フレーム間の物体移動量から推定した物体速度に基づいて、物体の探索範囲を決める。これにより、フレーム全体を探索するのに比べて追跡処理に掛かる計算量を減らすことができる。次に「追跡処理」では、この探索範囲内においてテンプレートマッチングを使うことで対象物体を検出する。

これらの2つの処理に基づいて得られる物体追跡システ

ムの消費エネルギーモデルは次の通りである。

$$E = (\alpha \cdot W \cdot H + (6 \cdot N \cdot M + 3) \left(\frac{2 \cdot v}{FR} + 1 \right)^2) \cdot E_{op} \cdot FR, \quad (1)$$

ここで、 α はメモリ読出しにかかる消費エネルギーとプロセッサで命令実行に必要なエネルギー (E_{op}) の比を表しており、この値はメモリやプロセッサの仕様に依存する。 $W \cdot H$ は入力画像のサイズ、 $N \cdot M$ はテンプレート画像のサイズ、 v は推定速度、 FR はフレームレートをそれぞれ表している。

動的フレームレート最適化手法は追跡処理の後に、最適フレームレートを適用する。この最適フレームレートは物体の動きに基づいたエネルギートレードオフによって決定される。追跡対象の物体が大きく動く場合は、追跡精度を維持するために高いフレームレートで処理する必要がある。しかしながら、実際には撮影中に物体が低速になる、または停止するケースもある、このような場合には探索範囲内から大きく移動することはないと予測されるため、常に高いフレームレートで追跡処理を実行することは非効率である。そこで、物体速度が遅い場合はフレームレート、つまり単位時間あたりの処理枚数を削減することで、「フレーム取得」および「追跡処理」の2つの処理を削減する。これにより、「追跡処理」における計算コストのみならず、「フレーム取得」のコストも考慮した消費エネルギー削減を達成することができる。

図1にフレームレート固定の従来の物体追跡処理手法と我々が提案している動的フレームレート最適化法の処理手順を示す。フレームレート固定の従来手法は追跡対象物体の推定速度から探索範囲を決定する。図1(a)に従来手法の処理の流れを示す。フレーム3において、一つ前のフレームで推定速度が大きくなったことにより、探索範囲が大きく広げられていることが分かる。同様に、1つ前のフレームで追跡速度が低下したことにより、フレーム4、フレーム5では探索範囲が減少している。

これに対して、動的フレームレート最適化手法は探索範囲に加えて、フレームレートの最適化も図る。ここで、最適フレームレートは、式(1)から E を最小とする FR を計算することで式(2)のように求められる。

$$FR_{min}(t_c) = 2 \cdot \sqrt{\frac{6 \cdot N \cdot M + 3}{\alpha \cdot W \cdot H + 6 \cdot N \cdot M + 3}} \cdot v(t_c). \quad (2)$$

図1(b)にこの手法の処理手順を示す。フレーム3までは、物体速度が速い間はフレームレートを最大値に設定するため、従来のフレームレート固定の手法と同様の追跡を行っている。もし、推定速度が遅くなった場合はフレームレートを減らすことを試みる。たとえば、図1(b)を見ると、フレーム4ではフレームレートを削減したことにより、フレームの処理をスキップしている。代わりに、フレーム5ではフレーム4での移動量も含めて探索範囲に収まるよ

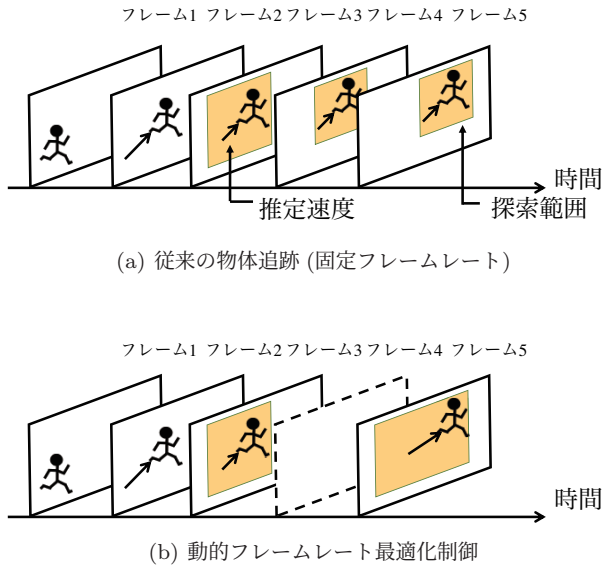


図 1 物体追跡の処理手順

うにするためにより大きく探索範囲を広げる。

2.2 追跡精度の改善

動的フレームレート最適化では、直前フレームで推定した速度が一定で続いていると仮定してフレームレートを決定する。しかしながら、実際には物体速度が急速に変化する場合があるため、上記のような単純な仮定に基づいてフレームレートを決定することは追跡精度に悪影響を与える可能性がある。推定速度が実際の速度よりも高い場合は、フレームレートを高く設定するため追跡精度を保つことができる。ただし、消費エネルギーを必要以上に消費するという欠点がある。これに対して推定速度が実際の速度を下回った場合、フレームレートを低く設定することで消費エネルギー削減効果が得ることができ、追跡精度の大幅な低下が生じる恐れがある。これは、推定速度に基づいて設定された探索範囲の外に物体が移動するためである。特に、推定速度が低い場合はフレームレートが低く設定されるとフレームの処理間隔が長くなるため、その間に物体の速度変化が生じることで追跡精度が悪化する。

ここでは、高い追跡精度を維持することを重視し、推定速度が実際の速度を下回った場合に追跡精度の低下を緩和するための改善について述べる。物体の速度が短時間に大幅に変化すると、推定速度と実際の速度との間に誤差が生じ、追跡精度に影響を与える。そこで、直近の速度変化が大きい場合は、フレームレート変更を停止させる。このような機能を実装するために、 NF と S_{th} と呼ぶ2つのパラメータを導入する。 NF は直近のフレーム枚数として定義されており、直近 NF 枚の速度変化を見るために用いる。 S_{th} はこの速度変化の大きさを判定するためのしきい値として定義される。図 2 に物体速度の時間変化および、 NF と S_{th} の設定例 ($NF = 5$ frames, $S_{th} = 20$ pixels/s) を示す。横軸は各フレームをキャプチャした時間、縦軸は各フ

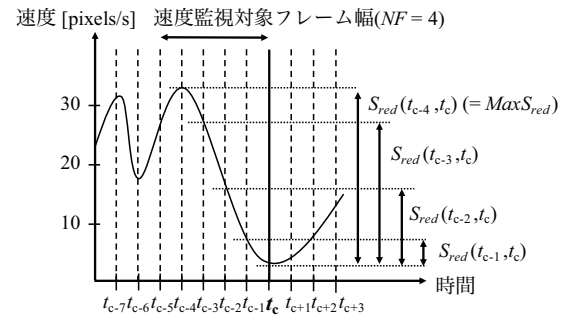


図 2 追跡精度改善のためのパラメータ

レームにおける物体の速度を表している。特に、現在フレームの時間を t_c で表す。ここで、 t_{i-n} と t_i の間の物体の平均速度の変化量 $S_{red}(t_{i-n}, t_i)$ は次式で計算される。

$$S_{red}(t_{i-n}, t_i) = \max\{v(t_{i-n}) - v(t_i), 0\}. \quad (3)$$

このとき、現在フレームから NF フレーム前までの間における最大の速度変化量は次のように定義される。

$$\text{Max}S_{red}(t_c, NF) = \max\{S_{red}(t_{c-j}, t_c) | j \in \mathbb{N}, j \leq NF\}. \quad (4)$$

図 2 の例では、 $\text{Max}S_{red}(t_c, NF)$ は $S_{red}(t_{c-4}, t_c)$ となる。もし、この $\text{Max}S_{red}(t_c, NF)$ が S_{th} を超える場合は、フレームレートを維持する。超えない場合はフレームレート速度が急激に変化しないとみなし、式 (2) に基づいてフレームレートの変更を実施する。

3. 撮影状況に基づいたエネルギー削減率改善

3.1 モチベーション

第 2 節では、これまでに我々が提案してきた動的フレームレート最適化手法について述べた。追跡精度を向上させるために NF と S_{th} の二つのパラメータを導入した。これらは、パラメータを変化させた予備実験において平均的に消費エネルギー削減率が改善される値を静的に与えている。しかしながら、実際には物体を追跡する状況は時々刻々と変化する。したがって、これらのパラメータの適切な値も状況に応じて変化すると考えられる。そこで、追跡処理中にこれらの NF と S_{th} の二つのパラメータを調節することを試みる。

3.2 パラメータ探索に基づく定量的解析

まず、どのように動的にパラメータを調節するべきかを検討するために、動画ごと (つまり、撮影状況ごと) の最適なパラメータ探索を行う。以下に示すいくつかの物体追跡モデルを用いて比較を行う。

FIXFR 比較基準のための、フレームレートを 30 fps で固定した一般的な物体追跡モデル。

ADAPTER 我々が提案している動的なフレームレート制御法であり、 (NF, S_{th}) を (5, 30) で固定したモデル。

ADAPTFR_OPT NF および S_{th} について網羅的にパラメータ探索を行い、各動画ごとに十分な追跡精度である 60%を超えた上で消費エネルギー削減率が最も高い NF と S_{th} のセットを用いた追跡モデル。なお、各パラメータの探索範囲は、 $0 \leq NF \leq 10$, $10 \leq S_{th} \leq 100$ で、この 100 通りの組み合わせから選択する。

入力動画として、物体追跡用のベンチマークである Tracker Benchmark [12] を用いる。ここで、各フレームごとの追跡精度はベンチマーク動画が用意している正解矩形と物体追跡システムが出力する矩形の重なり率とし、次式によって求められる。

$$Accuracy = \frac{area(A) \cap area(Q)}{area(A) \cup area(Q)} \times 100 [\%], \quad (5)$$

評価に用いた全フレームにおける追跡精度の平均値を評価指標として用いる。また、消費エネルギーは式 (1) で計算される。 α はメモリやプロセッサの仕様に依存する値であり、ここでは実際のデバイスの仕様 [13], [14] に基づいて $\alpha = 5$ であると仮定する。

図 3, 図 4 に各ベンチマークに対する追跡精度と、消費エネルギーを示す。各バーに **ADAPTFR_OPT** において、選択された (NF, S_{th}) のパラメータ値を付す。一般的な追跡モデルであるフレームレート固定の **FIXFR** と比較して、**ADAPTFR** および、**ADAPTFR_OPT** は大幅な追跡精度の低下無しに、消費エネルギーの削減を達成することができている。特に、**ADAPTFR_OPT** は **ADAPTFR** と比較して、最大で 67.1%(*FaceOcc1*), 平均で 32.3%の消費エネルギーの削減余地がある。したがってこの結果から、動画ごとに最適な NF と S_{th} を与えることができれば、消費エネルギーをより削減することができることが分かる。たとえば、 $(NF, S_{th})=(1, 10)$ または $(1, 60)$ の場合に、4つのベンチマーク (*Dog1*, *Doll*, *Dudek*, *FaceOcc1*) に対して最適なパラメータとなっている。

3.3 動的なパラメータ選択

第 3.2 節では、動画ごとに最適な NF と S_{th} があることが分かった。加えて、実際の物体追跡では撮影状況が時々刻々と変化していく。したがって、最適な NF と S_{th} の組み合わせも変化していくと考えられる。そこで、消費エネルギー削減率の改善のために、物体追跡処理をしながら、動的に NF と S_{th} を切り替える手法を提案する。具体的には、撮影状況の 1 つとして物体の移動速度から NF と S_{th} のパラメータ選択を行う。

アルゴリズム 1 に、提案手法の疑似コードを示す。1–6 行目はフレームレート固定の一般的な物体追跡処理と同様であり、1) フレームの読み出し、2) 物体速度に応じた探索範囲の設定、3) 探索範囲内でテンプレートマッチング、4) 物体速度の推定を順次実行する。続いて 7–11 行目では、本

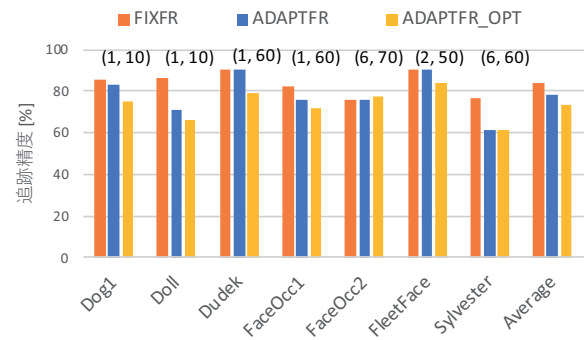


図 3 追跡精度予備評価

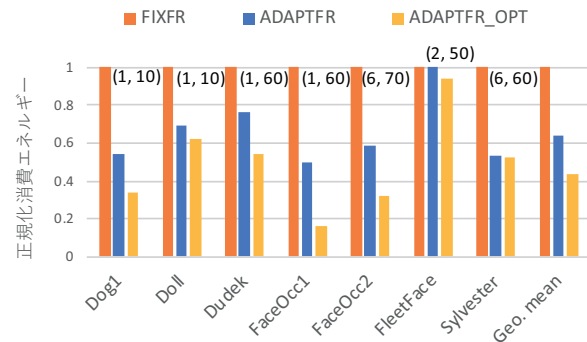


図 4 消費エネルギー予備評価

節で提案したパラメータ選択を行う。パラメータ切り替えのための速度しきい値 $Param_{th}$ を与え、現在の物体速度 $v[t_c]$ が閾値を越えるかどうかによって NF と S_{th} の値を変更する。ここでは、第 3.2 節でより多くの動画画像で最適なパラメータとなった $(NF, S_{th})=(1, 10)$ または $(1, 60)$ を物体の移動速度に応じて変更する。その後、12–16 行目で第 2.2 節で述べたフレームレート制御を行う。選択された NF と S_{th} に基づき、フレームレート変更の可否を判断する。もし、速度変化が著しい場合はフレームレートを現在の値に固定する (12–13 行目)。反対に、継続して一定の速度変化に収まる場合は物体の移動が急激に変わることはないと判断し、フレームレート変更を許容する (14–15 行目)。

4. 提案手法の定量的評価

4.1 評価環境

本評価では、第 3.3 節で利用した物体追跡のモデル **FIXFR** および **ADAPTFR** に加えて、本論文での提案モデルとして **ADAPTFR_{Param}** を用いる。このモデルは **ADAPTFR** に加えて、物体速度に応じて $(NF, S_{th})=(1, 10)$ または $(1, 60)$ に動的に設定する。これらのモデルを評価するために、OpenCV ライブラリを用いてシミュレータを実装した。本評価では、物体追跡向けベンチマークである Tracker Benchmark [12] から 7 つの動画画像を用い、その先頭から 600 フレームを評価対象とした。なお、これらの動画画像のフレームレートは明記されていないため、一般的な値である 30fps であると仮定する。また、テンプレートマッチン

アルゴリズム 1 物体の速度に応じたフレームレート制御

Require: NF, S_{th}

Ensure: $coordinate[i]$

```

1: while SystemIsRunning() do
2:    $t_c \leftarrow t_i$ 
3:    $frame[i] \leftarrow TakeFrame(t_c)$ 
4:    $search\_area \leftarrow DecisionSearchArea(coordinate[i - 1],$ 
      $v[t_{c-1}])$ 
5:    $coordinate[i] \leftarrow Tracking(frame[i], search\_area)$ 
6:    $v[t_c] \leftarrow Prediction(coordinate, FR_{current})$ 
7:   if  $v[t_c] > Param_{th}$  then
8:      $(NF, S_{th}) \leftarrow (NF_1, S_{th1})$ 
9:   else
10:     $(NF, S_{th}) \leftarrow (NF_2, S_{th2})$ 
11:   end if
12:   if  $MaxS_{red}(t_c, NF) > S_{th}$  then
13:      $FR_{next} \leftarrow FR_{current}$ 
14:   else
15:      $FR_{next} \leftarrow FR_{min}(t_c)$ 
16:   end if
17:    $i \leftarrow i + 1$ 
18: end while

```

グのためのテンプレート画像はこのベンチマークから正解矩形を抽出したものをを用いる。追跡精度および消費エネルギーは上述した、式 (5)、式 (1) によって計算される。

4.2 評価結果

追跡精度および、消費エネルギーの評価結果を図 5 および図 6 にそれぞれ示す。横軸はそれぞれ各動画像、縦軸は追跡精度もしくは、 $FIXFR$ で正規化された消費エネルギーを表している。 $ADAPTFR_{Param}$ において、パラメータ選択のしきい値である $Param_{th}$ を 10, 20, 30 pixels/s の 3 つのケースについて評価する。これらの結果を見ると、 $ADAPTFR_{Param}$ は $ADAPTFR$ と比べて、 $Param_{th}$ が 20 pixels/s を越える場合、*FleetFace*, *Sylvester* を除く 5 つのベンチマークにおいて追跡精度の大幅な低下無しに、消費エネルギーをさらに削減できていることが分かる。消費エネルギー削減が得られなかった 2 つのうち、*FleetFace* は速度変化が常に大きい動画像であるため、フレームレートを変更する機会が無く、 $FIXFR$ と同様の結果が得られている。*Sylvester* では、 $ADAPTFR_{Param}$ は $ADAPTFR$ よりも消費エネルギーが増加しているが、代わりに追跡精度が向上していることが分かる。 $ADAPTFR_{Param}$ は $FIXFR$ と比較すると、最大で 64.3% (*FaceOcc1*)、平均で 45.0% の消費エネルギー削減を達成した。 $ADAPTFR$ と比較すると、提案手法は平均で約 13.2% の消費エネルギー削減率を改善した。これらの結果から、物体の移動速度といった撮影状況に応じて NF と S_{th} の 2 つのパラメータ

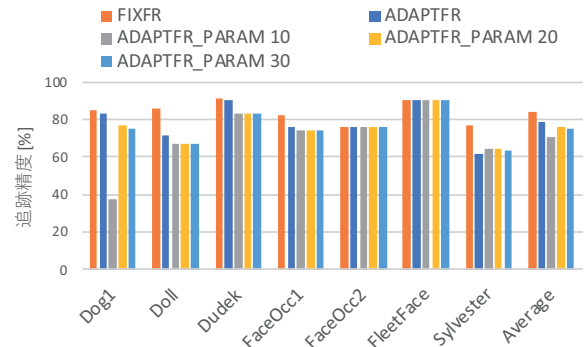


図 5 追跡精度評価結果

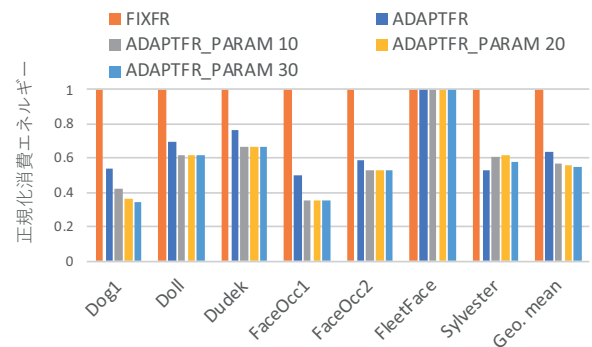


図 6 消費エネルギー評価結果

の設定を動的に変更することで追跡精度または消費エネルギーの改善を達成することが分かった。

5. まとめ

本論文では、動的フレームレート制御法の消費エネルギー削減率改善手法について述べた。これまで我々が提案してきた動的フレームレート制御法では、どのような入力動画像においても NF および、 S_{th} のパラメータ値を固定して追跡を行ってきた。しかしながら、入力される動画像中の物体の動きといった撮影状況に対してより良いフレームレートを設定することができない。そこで、この問題を解決するために、提案手法はこれらの 2 つのパラメータの値を入力動画像のシーンに合わせて動的に設定する手法を提案した。この提案手法を適用した結果、大幅な追跡精度の悪化無しに、フレームレート固定の従来手法よりも最大で 64.3%、平均で 45.0% の消費エネルギーを削減を達成した。また、これまで我々が提案してきた動的フレームレート制御法と比較して平均で 13.2% の消費エネルギー削減が見込めることが分かった。本論文の評価はシミュレーションに留まっているため、今後は実システム上への実装および評価を行う。

謝辞 本研究は JSPS 科研費 JP17K19984 の助成を受けたものです。

参考文献

- [1] Cottini, N., Gottardi, M., Massari, N., Passerone, R. and Smilansky, Z.: A 33 μ W 42 GOPS/W 64x64 pixel vision sensor with dynamic background subtraction for scene interpretation, *Proceedings of the 14th ACM/IEEE international symposium on Low power electronics and design - ISLPED '09*, p. 315 (2012).
- [2] Hanson, S. and Sylvester, D.: A 0.45-0.7V sub-microwatt CMOS image sensor for ultra-low power applications, *VLSI Circuits, 2009 Symposium on*, pp. 176–177 (2009).
- [3] LiKamWa, R., Priyantha, B., Philipose, M., Zhong, L. and Bahl, P.: Energy characterization and optimization of image sensing toward continuous mobile vision, *Proceeding of the 11th annual international conference on Mobile systems, applications, and services - MobiSys '13*, p. 69 (2013).
- [4] McIlrath, L. G.: A low-power low-noise ultrawide-dynamic-range CMOS imager with pixel-parallel A/D conversion, *IEEE Journal of Solid-State Circuits*, Vol. 36, No. 5, pp. 846–853 (2001).
- [5] Couniot, N., de Streel, G., Botman, F., Lusala, A. K., Flandre, D. and Bol, D.: A 65 nm 0.5 V DPS CMOS Image Sensor With 17 pJ/Frame.Pixel and 42 dB Dynamic Range for Ultra-Low-Power SoCs, *IEEE Journal of Solid-State Circuits*, Vol. 50, No. 10, pp. 2419–2430 (2015).
- [6] Tanabe, J., Toru, S., Yamada, Y., Watanabe, T., Okumura, M., Nishiyama, M., Nomura, T., Oma, K., Sato, N., Banno, M., Hayashi, H. and Miyamori, T.: A 1.9TOPS and 564GOPS/W heterogeneous multicore SoC with color-based object classification accelerator for image-recognition applications, *2015 IEEE International Solid-State Circuits Conference*, pp. 1–3 (2015).
- [7] Usui, H., Tanabe, J., Sano, T., Xu, H. and Miyamori, T.: An evaluation of an energy efficient many-core SoC with parallelized face detection, *2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 311–316 (2014).
- [8] Oh, J., Kim, G., Park, J., Hong, I., Lee, S. and Yoo, H. J.: A 320mW 342GOPS Real-Time Moving Object Recognition Processor for HD 720p Video Streams , *2012 IEEE International Solid-State Circuits Conference*, pp. 220–222 (2012).
- [9] Sasagawa, Y. and Mori, A.: High-level Video Analytics PC Subsystem Using SoC With Heterogeneous Multi-core Architecture, *IEEE Journal of Solid-State Circuits*, Vol. 51, No. 4, pp. 1051–1059 (2016).
- [10] Inoue, Y., Ono, T. and Inoue, K.: Adaptive Frame-Rate Optimization for Energy-Efficient Object Tracking, *International Conference on Image Processing, Computer Vision, Pattern Recognition*, pp. 1–7 (2016).
- [11] Inoue, Y., Ono, T. and Inoue, K.: Real-time Frame-Rate Control for Energy-Efficient On-Line Object Tracking, *The Institute of Electronics, Information and Communication Engineers Transactions on Fundamentals of Electronics, Communications and Computer Sciences*. to be published.
- [12] Wu, Y., Lim, J. and Yang, M.-H.: Online object tracking: a benchmark, *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2411–2418 (2013).
- [13] Kikuchi, Y., Takahashi, M., Maeda, T., Fukuda, M., Koshio, Y., Hara, H., Arakida, H., Yamamoto, H., Hagiwara, Y., Fujita, T., Watanabe, M., Ezawa, H., Shimazawa, T., Ohara, Y., Miyamori, T., Hamada, M., Takahashi, M. and Oowaki, Y.: A 40 nm 222 mW H.264 Full-HD decoding, 25 power domains, 14-core application processor with x512b stacked DRAM, *IEEE Journal of Solid-State Circuits*, Vol. 46, No. 1, pp. 32–41 (2011).
- [14] Tanabe, Y., Sumiyoshi, M., Nishiyama, M., Yamazaki, I., Fujii, S., Kimura, K., Aoyama, T., Banno, M., Hayashi, H. and Miyamori, T.: A 464GOPS 620GOPS/W heterogeneous multi-core SoC for image-recognition applications, *2012 IEEE International Solid-State Circuits Conference*, pp. 222–223 (2012).