Vol.2016-CVIM-201 No.7
2016/3/3

部分領域の追跡破綻検出を用いた追跡対象の変化と類似物体に頑健な動画像追跡

西村 景汰[1,a)]

概要：動画像から特定の対象を追跡する問題である visual tracking は，コンピュータビジョンの分野で重要な問題であり，さまざまな応用が期待されている [1]．visual tracking の研究は盛んに行われており，その研究課題は追跡対象の姿勢の変化，隠れ，類似背景，照明変化，ノイズ，急激な動きが発生しても追跡が破綻しない頑健な，かつ実時間で処理可能なアルゴリズムの開発とされている．Zhang らはシンプルかつ対象の変化に対して頑健な Compressive Tracking（CT） [2] を提案し，Zhu らは対象を部分領域に分け，その部分領域を独立して追跡し，その結果を統合することで，隠れに対しての頑健性を上げるように CT を改良したアルゴリズム [3] を提案した．[3] では，追跡している部分領域と類似する見え方を持つ領域が現れ，かつ対象の見え方が変化をしている場合，部分領域が類似する領域の追跡を始めてしまい，長時間の追跡を行うと正しく追跡が行えないという問題点を抱えている．本研究では，Kalal らにより提案された追跡破綻検出アルゴリズム [4] を用いて，追跡が破綻した部分領域の追跡を打ち切り，部分領域を再生成する手法を用いることで上記の問題点を解決し，一部の動画において大幅な追跡性能向上を確認した．

1. はじめに

動画像から特定の対象を追跡する問題である visual tracking は，コンピュータビジョンの分野で重要な問題であり，さまざまな応用が期待されている．店舗に設置されている防犯カメラから人物を追跡して，自動防犯システムへの応用や，客の移動情報から，客の商品に対しての関心を予測するマーケティング支援システムへの応用，あるいは人間とコンピュータのインタラクションに役立てるといった，幅広い応用が期待されている [1]．

visual tracking の研究は盛んに行われており，その研究課題は追跡対象の姿勢の変化，隠れ，類似背景，照明変化，ノイズ，急激な動きが発生しても追跡が破綻しない頑健な，かつ実時間で処理可能なアルゴリズムの開発とされている．本研究でも，このような対象の変化に対して頑健な visual tracking アルゴリズムを開発することを目的とする．

Zhang らはシンプルかつ対象の変化に対して頑健な Compressive Tracking（CT） [2] を提案した．Zhu らは対象を部分領域に分け，その部分領域を独立して追跡し，その結果を統合することで，隠れに対しての頑健性を上げるように CT を改良したアルゴリズム [3] を提案した．[3] では，追跡している部分領域と類似する見え方を持つ領域が現れ，かつ対象の見え方が変化をしている場合，部分領域が類似する領域の追跡を始めてしまい，長時間の追跡を行うと正しく追跡が行えないという問題点を抱えている．本研究では，Kalal らにより提案された追跡破綻検出アルゴリズム [4] を用いて，追跡が破綻した部分領域の追跡を打ち切り，部分領域を再生成する手法を用いることで上記の問題点を解決し，一部の動画において大幅な追跡性能向上を確認した．

本報告の構成を示す．2 章，3 章では先行研究 [2]，[3] を概説する．4 章で [3] の問題点を指摘し，5 章で追跡破綻検出 [4] を概説した後，6 章で本研究での提案手法を説明する．7 章で提案手法を実装し，追跡性能の比較を行った結果を述べる．8 章でまとめと今後の課題を述べる．

2. Compressive Tracking

本章では，先行研究である [2] を概説する．

2.1 Random Projection

CT では，高次元のベクトルをお互いの距離関係をを保ったまま，低次元に射影する手段として，random projection を用いている．以下，具体的な射影方法を示す．d 次元空間から k 次元空間（$d \gg k$）への射影行列 $R \in \mathbb{R}^{k \times d}$ を疎な行列になるようにランダムに生成する．具体的には，(1) にしたがって，行列 R の (i, j) 成分を決定する．この R は，

1 九州工業大学
Kyushu Institute of Technology
a) o676025k@mail.kyutech.jp

© 2016 Information Processing Society of Japan

追跡アルゴリズムの実行前に 1 度だけ生成する．

$$r_{ij} = \sqrt{s} \times \begin{cases} 0 & (\text{with probability } 1 - \frac{1}{s}) \\ +1 & (\text{with probability } \frac{1}{2s}) \\ -1 & (\text{with probability } \frac{1}{2s}) \end{cases} \quad (1)$$

CT では，$s = \frac{d}{4}$ としている．Johnson-Rindenstrauss の補題に基づき，線形写像 R は高い確率で射影前後で点同士の距離関係がおおよそ保存される線形変換行列となることが知られているため，変換後の低次元ベクトルは，変換前の高次元ベクトルの情報をおおよそ保つ．

2.2 対象からの特徴抽出

CT では，追跡対象を 2 次元の矩形領域で表現し，フレーム画像から矩形領域で切り出される画像に対して，次のような形で矩形領域から特徴抽出をしている．まず，1 ピクセル ×1 ピクセルから追跡対象の矩形のサイズまで，1 ピクセル単位で矩形のサイズを変化させながら矩形領域で切り出した画像に対して矩形をたたみこみ，矩形たたみこみ画像をたたみこんだ矩形の数だけ生成する．それぞれの画像を 1 つの列ベクトルとし，すべての列ベクトルを縦に連結するとおよそ 10^6 次元から 10^{10} 次元ベクトルとなる．このおよそ 10^6 次元から 10^{10} 次元のベクトルを random projection により 50 次元に圧縮し，それを切り出した矩形領域の特徴ベクトルとする．

2.3 対象モデル・背景モデル

CT では，フレームから切り出される矩形領域が生成される母集団として，対象クラスと背景クラスを仮定し，その特徴ベクトルの確率分布（以後，対象と背景のモデルと呼ぶ）を推定して，フレームから切り出した矩形領域の対象らしさを計算している．以下，その対象と背景のモデルの構築の仕方を述べる．

対象と背景のモデルを推定するためのサンプルとして，そのフレームで追跡対象がいた位置の近くから（一定半径以内）一定枚数，追跡対象を表す矩形の大きさで矩形領域を切り出し，それを対象サンプルとする．また，そのフレームで対象がいた位置の遠くから一定枚数（一定半径以上一定半径以内），追跡対象を表す矩形の大きさで矩形領域を切り出し，それを背景サンプルとする．

対象クラスの特徴ベクトルの分布，背景クラスの特徴ベクトルの分布を，それぞれが異なったパラメータを持つ，次元ごとに独立な正規分布であると仮定し，それぞれ対象モデル，背景モデルとする．すなわち，

$$p(\boldsymbol{v}|+1) = \prod_{i=1}^{k} p(v_i|+1) \quad (2)$$

$$p(\boldsymbol{v}|-1) = \prod_{i=1}^{k} p(v_i|-1) \quad (3)$$

$$p(v_i|+1) \sim \mathcal{N}(\mu_i^{+1}, (\sigma_i^{+1})^2) \quad (4)$$

$$p(v_i|-1) \sim \mathcal{N}(\mu_i^{-1}, (\sigma_i^{-1})^2) \quad (5)$$

ただし，$\boldsymbol{v}$ はサンプルの特徴ベクトルを表し v_i は $\boldsymbol{v}$ の第 i 成分を表すものとする．+1 は対象クラス，−1 は背景クラスを表す．このパラメータである $\mu_\cdot$ と $\sigma_\cdot$ を毎フレーム，次の式に従って更新する．

$$\mu_i^{+1} \leftarrow \lambda\mu_i^{+1} + (1-\lambda)\overline{\mu_i^{+1}} \quad (6)$$

$$(\sigma_i^{+1})^2 \leftarrow \lambda(\sigma_i^{+1})^2 + (1-\lambda)(\overline{\sigma_i})^2 + \lambda(1-\lambda)(\mu_i^{+1} - \overline{\mu_i^{+1}})^2 \quad (7)$$

ただし，

$\overline{\mu_i^{+1}}$ 対象サンプルから求めた特徴ベクトルの平均の第 i 成分

$\overline{\sigma_i^{+1}}$ 対象サンプルから求めた特徴ベクトル標準偏差の第 i 成分

λ 学習係数 $0 \le \lambda \le 1$ （CT では $\lambda = 0.85$ としている）

+1 を −1，対象を背景と読み替え，背景モデルのパラメータも更新する．

2.4 CT のアルゴリズム

CT では，対象の検出および，追跡に用いるモデルの更新を各フレームで行うことにより，対象の追跡を実現している．以下，前者を検出ステップ，後者を更新ステップと呼ぶ．

2.4.1 検出ステップ

そのフレームにおいて，対象がいるとされる画像上の矩形領域を推定するステップである．

まず，そのフレームにおける画像において，前のフレームの対象位置の近くから（一定半径以内で），追跡対象を表す矩形の大きさで矩形領域を切り出し，各々から特徴抽出を行う．サンプルの各々の特徴ベクトル $\boldsymbol{v}$ につき，対象らしさ $L(\boldsymbol{v})$ を次の式で計算する．

$$L(\boldsymbol{v}) = \ln \frac{p(\boldsymbol{v}|+1)}{p(\boldsymbol{v}|-1)} = \ln p(\boldsymbol{v}|+1) - \ln p(\boldsymbol{v}|-1) \quad (8)$$

ただし，$p(\boldsymbol{v}|+1)$ はサンプルの対象に対する尤度，$p(\boldsymbol{v}|-1)$ はサンプルの背景に対する尤度を示す．この対象らしさ $L(\boldsymbol{v})$ を最大にするサンプルを選択し，それが抽出された位置をそのフレームにおける対象の位置とする．

2.4.2 更新ステップ

2.3 節で記したように，対象サンプルと背景サンプルを抽出し，モデルパラメータを更新する．

3. 部分領域モデルと CT の融合

本章では，[3] を概説する．

CT では，対象が隠れた場合には対象を正しく検出できなかったり，モデルパラメータが追跡したい対象を表現し

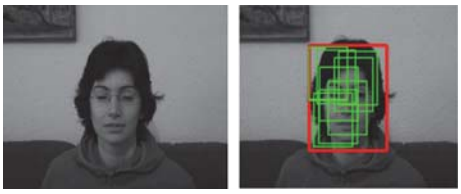

図 1 対象領域からの部分領域の抽出.
左：[5] の face sequence の 0 フレーム目をグレースケールに変換したもの
右：[3] の図 2 の左側を抜粋
Fig. 1 SubRegion Extraction.
Left : from [5](0 frame in face sequence) Right : Left side of Fig.2 from [3]

たものではなくなったりする（モデルのドリフト）[3]．そこで，Zhu ら [3] は，対象を部分領域に分け，その部分領域を独立に追跡し，隠れが起きていないと考えられる部分領域のみの追跡結果を用いて最終的な対象の位置を推定する方法で部分的な隠れによるモデルのドリフトを解決した．以下に具体的なアルゴリズムを示す（以下，[3] で提案されているアルゴリズムを CTSubRegion と呼ぶ）．

3.1 部分領域の分割と追跡

最初のフレームにおいて，対象領域から部分領域を抽出する．具体的には，対象領域の矩形に収まる範囲で，あらかじめ定めた幅，高さで矩形を 20〜30 個抽出する（**図 1**）．また，それぞれの部分領域について，対象全体の領域を表している矩形の左上座標から，部分領域を表している矩形の左上座標までの距離（矩形オフセット）を記録する．1 フレームごとに，その部分領域を独立に CT のアルゴリズムで追跡を行う．CT では，特徴ベクトルの次元数を 50 次元としているが，CTSubRegion では計算時間を削減するため，特徴ベクトルの次元数は 10 次元としている．

3.2 結果の統合

各部分領域について，式 8 に示した対象らしさが上位であった領域（上位の基準を，上位 1/4 までから上位 1/2 までの範囲で事前に定める）を結果の統合に用いる（本論文では，有効部分領域と呼ぶ）．この有効部分領域は，ほとんどの場合隠れていない部分領域となる．有効部分領域の 1 つ 1 つについて，事前に登録した矩形オフセットを用いて対象全体を表す矩形の左上座標の候補を計算する．その後，その左上座標から半径一定以内のすべての点に 1 の投票を行う．投票面では,x, y 座標はそれぞれ 0 から 1 の範囲内に正規化されているものとし，投票半径は 0.05 としている．

その後，投票数が最大になった点を探索する．具体的には，各投票中心点を初期点として mean-shift のアルゴリズムを実行し（カーネルには投票に用いたものと同じものを用いる），投票数が極大になった点を求める．この極大値のうち，最大のものに対応する点を，そのフレームにおける対象全体を表す矩形の左上座標とする．

3.3 部分領域モデルの更新

各々の部分領域が追跡に用いている対象モデルと背景モデルの更新を，以下の条件（本論文ではモデル更新条件と呼ぶ）を満たす部分領域については学習係数を小さくして，満たさない部分領域については学習係数を大きくして行う．

- 対象らしさが閾値を超える部分領域
- そのフレームで対象全体がいるとされた領域の左上座標と，部分領域の左上座標からオフセットを減算した位置の距離が閾値未満である部分領域

上記の条件を満たさない部分領域はその追跡位置の情報を信頼できないと考えられるため，上記のような部分領域のモデルを更新しないことで，モデルのドリフトを防ぐ．

4. CTSubRegion の問題点

[3] では，追跡に用いている部分領域と類似する見え方を持つ領域が現れ，かつ追跡対象の見え方が変化をしている場合，部分領域が類似する領域の追跡を始めてしまい，長時間の追跡を行うと正しく追跡が行えないという問題点を抱えている．本研究では，Kalal らにより提案された追跡破綻検出アルゴリズム [4] を用いて，追跡が破綻した部分領域の追跡を打ち切り，部分領域を再生成する手法を用いることで上記の問題点を解決することを試みる．

5. 追跡破綻検出

ここでは，本研究にて採用する追跡破綻検出 [4] について述べる．

現在時刻を t とし，それより $k(>0)$ 時間前の時刻 $t-k$ において，追跡している対象の推定位置が $\boldsymbol{x_{t-k}}$ であったとする．このとき，現在時刻 t において追跡している対象の推定位置 $\boldsymbol{x_t}$ から，時間軸を逆向きにして（動画を逆再生して）時間 k だけ，追跡アルゴリズムを実行し，その結果得られた時刻 $t-k$ における対象の推定位置が $\overline{\boldsymbol{x_{t-k}}}$ であったとする．このとき，$|\boldsymbol{x_{t-k}} - \overline{\boldsymbol{x_{t-k}}}|$ を計算し，それがあらかじめ定めた閾値以上ならば，時刻 t において追跡が破綻したものとみなす．

図 2 に，追跡破綻検出の例として，時刻 $t-k$ と時刻 t において，車の側面前方のミラー（1），前輪（2）を追跡している様子を示している．黄色の点が時刻 $t-k$ において追跡している点 $\boldsymbol{x_{t-k}}$，青色の点が時刻 t において追跡している点 $\boldsymbol{x_t}$，赤色の点が時刻 t から時間軸を逆向きにして時間 t だけ追跡した結果 $\overline{\boldsymbol{x_{t-k}}}$ を示している．1 の場合は，黄色の点と赤色の点の距離である $|\boldsymbol{x_{t-k}} - \overline{\boldsymbol{x_{t-k}}}|$ が小さく，2 の場合は $|\boldsymbol{x_{t-k}} - \overline{\boldsymbol{x_{t-k}}}|$ が大きい．$|\boldsymbol{x_{t-k}} - \overline{\boldsymbol{x_{t-k}}}|$ について適当な閾値処理を行い，1 の場合は追跡が破綻していない，2 の場合は追跡が破綻しているとみなす．

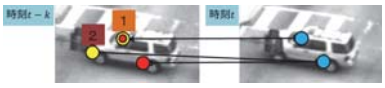

図 2 追跡破綻検出．[4] の図 1 に著者が手を加えて作成
Fig. 2 Tracking Failure Detection. From Fig.1 in [4] and some modify.

Algorithm 1 提案アルゴリズム

Input: t-th video frame F
Output: target object region R in t-th video frame
  for all $R_i \in subRegions$ **do**
    $R_i \leftarrow CT(F, R_i)$
  end for
  $R \leftarrow combine(subRegions)$
  for all $R_i \in subRegions$ **do**
    if $trackFailure(Ri)$ **then**
      $R_i \leftarrow generateSubRegion(R)$
    else if $confidence(R_i) > confidenceThreshold$ **and** $distance(topleft(R), topleft(R_i) - offset(R_i)) < distanceThreshold$ **then**
      $updateRegionModel(R_i, \lambda_{low})$
    else
      $updateRegionModel(R_i, \lambda_{high})$
    end if
  end for

6. 提案手法

6.1 提案手法の概要

CTSubRegion[3] の，追跡に用いている部分領域と類似する見え方を持つ領域に弱いという欠点を克服するため，追跡破綻検出 [4] を用いて部分領域の動的な選別を行い，類似物体が現れても安定して追跡ができるアルゴリズムを提案する．これにより，対象の変化と類似物体の出現に対して頑健なアルゴリズムとできることを期待する．

6.2 追跡破綻検出を用いた部分領域の選別

CTSubRegion[3] において，対象全体の位置を推定した後，各部分領域について追跡破綻検出を適用する．その結果追跡が破綻しているとされた部分領域については，その部分領域の追跡を打ち切り，そのフレームで対象がいるとされた領域からランダムに部分領域を 1 つ生成し（大きさ，位置の決め方は初期フレームでの部分領域の抽出のアルゴリズムに準ずる），代わりにその部分領域を追跡するようにする．**図 3** の左側では，手のひらから見て背景にあたる手首を追跡し始めた部分領域を赤色の矩形で，そのフレームにおける手のひらの推定位置を水色の矩形で示している．図 3 の左側で，赤色の矩形は追跡破綻検出で追跡が破綻されたとみなされるが，その後，赤色の矩形の追跡を打ち切り，水色の矩形内で新たに追跡する矩形をランダムに生成する（図 3 の右側において，赤色で示している）．

以上に述べたアルゴリズムを Algorithm 1 に示す．

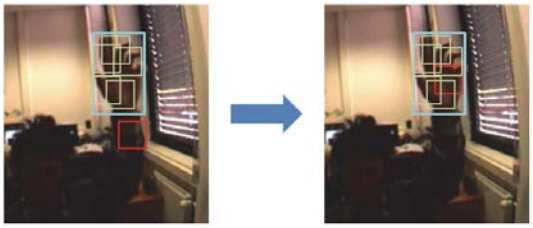

図 3 部分領域の再生成．[6] で公開されている動画 “hand” の 1 フレームを抜粋し，枠を人手で書き加えた
Fig. 3 Regenerate SubRegion. One frame in “hand” sequence from [6], rectangle is written by author.

# of subregions	20
# of valid subregions / # of subregions	20
# of backward tracking frames	2
tracking failure distance threshold	3

表 1 アルゴリズムのパラメータ
Table 1 Parameter of Algorithm.

7. 実験

6 章にて提案したアルゴリズムを実装し，性能評価実験を行った．動画は [6] で公開されているのものを用いた．パラメータは**表 1** のとおりとした．

ここに明記しなかったものについては，原則として [2] ないし [3] のものを用いる．[3] 中に明確な記述のなかった，部分領域ごとの学習係数については，部分領域モデル更新条件を満たした場合の学習係数を 0.85，満たさなかった場合の学習係数を 1 とした．CTSubRegion におけるモデル更新条件について，対象らしさの閾値を 0，位置誤差の閾値を対象全体を表す矩形の対角線の長さの 0.1 倍とした．評価の基準として，[2] で用いられている 1 動画中の成功率（Success Rate）を用いる．各フレームにおいて，(9) で定義される値が 0.5 以上であるとき，そのフレームは追跡が成功したとみなす．ある動画について，追跡が成功したフレーム数の，動画の全フレーム数に対する割合を成功率と定義する．

$$\frac{area(R_T \cap R_G)}{area(R_T \cup R_G)} \tag{9}$$

ただし，

area(.)　矩形領域. の面積
R_T　追跡アルゴリズムの出力したそのフレームの対象矩形
R_G　そのフレームにおける対象の外接長方形（Ground Truth）．[6] で公開されているものを用いた．

表 2 に全動画における成功率とその平均 *1，および平均 FPS *2 を示す．CT に比較して大きく改善の見られた動画（hand）のスクリーンショットを**図 4** に示す．動画の途中

*1 各動画の成功率のその動画を構成するフレーム数で重みづけした算術平均
*2 各動画の平均 FPS のその動画を構成するフレーム数で重みづけした調和平均

動画名	CT	CTSubRegion	提案手法
bicycle	0.626	0.332	0.479
can	0.758	0.632	0.627
dinosaur	0.164	0.168	0.227
gymnastics	0.176	0.210	0.159
hand	0.151	0.395	0.790
hand2	0.152	0.095	0.164
torus	0.108	0.117	0.066
平均成功率	0.294	0.265	0.348
平均 FPS	65.815	16.326	5.173

表 2 追跡アルゴリズムの成功率と平均 FPS
Table 2 Success Rate and Average FPS of Tracking Algorithm.

に追跡している手と部分的に類似する領域が現れ（手の陰になった部分と，衣服の黒い部分が類似している），CT では追跡が破綻しているが，提案手法は追跡が破綻することなく手を追跡している．この動画では，追跡対象である手の見た目が大きく変化し，かつ類似する領域が現れており，今回提案した部分領域の動的な選択が有効に機能していると考えられる．その他の動画については，CT に比較して性能が上がるものも下がるものもあった．

上記実験とは別に，CTSubRegion と提案手法で [6] の動画に対して追跡アルゴリズムを実行し，追跡を行っている部分領域を可視化した．**図 5** にそれを示す．凡例は以下のとおりである．

緑色矩形 有効部分領域
紫色矩形 有効部分領域でない部分領域
赤色楕円 有効部分領域により投票を行う領域の境界に引いた楕円 (楕円の中心は投票中心，すなわち矩形左上座標 – 矩形オフセット)
青色矩形 追跡対象の位置

図 5 から，CTSubRegion では追跡を行っている部分領域と似た領域（手首）を追跡している有効部分領域が多く現れ，有効部分領域の投票点が手首を囲む矩形の左上座標に近くなり，手の追跡に失敗しているが，提案手法では，手首を追跡している部分領域はなく，すべての部分領域が手を追跡している．CTSubRegion の結果から，提案手法においても途中で手首を追跡する部分領域が現れた可能性は大きいが，そのような部分領域が現れても，追跡破綻検出によりその部分領域の追跡が打ち切られるため，手首を追跡する部分領域が少なくなり，正しく手を追跡したものと考えられる．

8. おわりに

対象の変化に対して頑健な追跡アルゴリズムの開発を行うため，対象の変化に強いとされる，部分領域モデルを用いる追跡アルゴリズムの 1 つである CTSubRegion に部分領域の追跡破綻検出を適用して，追跡対象と部分的に類似する領域に対する頑健性の高いアルゴリズムを提案した．

図 4 改善の見られた動画 (左が CT，中央が CTSubRegion，右が提案手法 *4)
Fig. 4 Movie which Performance Inproved (Left : CT, Center : CTSubRegion, Right : Our Algorithm).

図 5 部分領域の可視化 (左が CTSubRegion，右が提案手法)
Fig. 5 Visualized Subregion (Left : CTSubRegion, Right : Our Algorithm).

今後の課題として，提案アルゴリズムの高速化があげられる．現状のアルゴリズムでは，[6] の動画に対して，平均 5.173FPS の処理速度となっており（Intel®Core^{TM} i7 CPU 920 @2.67GHz, 3GB RAM），15FPS の USB カメラからの動画をリアルタイムに処理できない．そのため，部分領域の追跡に用いている CT の高速化（[7] などが提案されている）を導入する，追跡破綻検出によらない部分領域の選択の提案などを用いて，アルゴリズムの高速化を図ることが課題となる．また，今回提案したアルゴリズムでは CTSubRegion で長所とされている隠れに対する頑健性が失われていると考えられる．そのため，部分領域モデルを用いた隠れ領域の推定アルゴリズムの開発を行い，部分領域の再生成戦略に応用する，あるいは部分領域モデルによらない，隠れに対処できる追跡アルゴリズムの開発が課題となる．

参考文献

[1] Lee, B. Y., Liew, L. H., Cheah, W. S. and Wang, Y. C.: Occlusion handling in videos object tracking: A survey, *IOP Conference Series: Earth and Environmental Science*, Vol. 18, No. 1, IOP Publishing, p. 012020 (2014).
[2] Zhang, K., Zhang, L. and Yang, M.-H.: Real-time compressive tracking, *Computer Vision–ECCV 2012*, Springer, pp. 864–877 (2012).
[3] Zhu, Q., Yan, J. and Deng, D.: Compressive tracking via oversaturated sub-region classifiers, *Computer Vision, IET*, Vol. 7, No. 6, pp. 448–455 (2013).

© 2016 Information Processing Society of Japan

[4] Kalal, Z., Mikolajczyk, K. and Matas, J.: Forward-backward error: Automatic detection of tracking failures, *Pattern Recognition (ICPR), 2010 20th International Conference on*, IEEE, pp. 2756–2759 (2010).

[5] : Fragtrack - Robust Fragments-based Tracking. `http://www.cs.technion.ac.il/~amita/fragtrack/fragtrack.htm`, 2016 年 2 月 4 日アクセス.

[6] : Research/LocalGlobalTracking - Visual Cognitive Systems Laboratory. `http://www.vicos.si/Research/LocalGlobalTracking`, 2016 年 1 月 27 日アクセス.

[7] Zhang, K., Zhang, L. and Yang, M.-H.: Fast compressive tracking, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 36, No. 10, pp. 2002–2015 (2014).

© 2016 Information Processing Society of Japan