

フレーム毎の特徴点抽出に基づく特徴点の追跡

市村 直幸

産業技術総合研究所 情報処理研究部門

〒 305-8568 つくば市梅園 1-1-1 中央第二

nic@ni.aist.go.jp, <http://staff.aist.go.jp/naoyuki.ichimura/>

あらまし 視野に出現する特徴点の追加を伴う特徴点の追跡は、視点の大きく移動する時系列画像解析のために必要である。本研究では、フレーム毎の特徴点抽出に基づく相関を用いた特徴点の追跡方法について考察する。この方法では、特徴点抽出結果に基づき対応候補を絞ることができるため、探索領域における枚挙的なマッチングを必要としない。そのため特徴点抽出を行う方法では、枚挙的なマッチングを行う方法に比べ、探索領域の大きさ、特徴点数、マッチングに用いる画像パッチの大きさの変化に対する計算時間の変化が非常に小さいという性質をもつことを示す。フレーム毎に特徴点抽出を行うため、新たな特徴点の追加は容易にできる。実画像を用いた実験結果と、パノラマ画像生成への応用を示す。

Feature Point Tracking Based on Feature Point Extraction in Every Frame

Naoyuki ICHIMURA

Information Technology Research Institute

National Institute of Advanced Industrial Science and Technology (AIST)

Tsukuba Central 2, 1-1-1, Umezono, Tsukuba, Ibaraki, 305-8568 Japan

nic@ni.aist.go.jp, <http://staff.aist.go.jp/naoyuki.ichimura/>

Abstract Feature point tracking with addition of feature points that enter view field is essential to analysis of image sequences with large view point change. This paper presents a tracking method using correlation based on feature point detection in every frame. Since feature point detection narrows down candidates of correspondence in search area, exhaustive search in search area is not required in the method. From this fact, I show that change in computational time of the method for changes in size of search area, number of feature points and size of image patch used in matching is quite smaller than one of the method using exhaustive search. Feature points that enter view field can easily be added to the tracker, because feature points are detected in every frame. Experiments for real images and an application to panoramic mosaic construction are included.

1 まえがき

特徴点の追跡は、時系列画像を用いたパノラマ画像生成、物体形状の推定、カメラの制御、動きの分割等において、特徴対応データを得るための一手段である [1]–[4]。特徴点を追跡するためには、フレーム間で画像データのマッチングを行う必要があり、数多くの方法が提案されている [5]–[10]。その中で、局所領域内での相関マッチングを用いる方法は、その簡明さのため広く用いられてきた [3][7]–[10]。この方法では、基本的に、初期フレームで特徴点抽出を行い、それらを引き続くフレームでの局所的な探索領域におけるマッチングにより追跡する。画像全体に対する処理が必要な特徴点抽出を初期フレームのみで行い、さらに、引き続くフレームでの対応点探索を、前フレームでの特徴点位置の近傍領域に限定することにより、計算量を低減している。しかし、この方法では初期フレームで抽出された特徴点のみしか追跡しないため、視点の移動に伴い視野に出現する特徴点を追跡することはできない。特徴点の出現や消失を考慮し、視点が大きく移動しても各々の視野での特徴対応データを得ることのできる追跡方法は、長期時系列画像解析のために必要である。

視野に出現する特徴点を追跡対象として逐次加えていく方法の一つとして、全てのフレームで特徴点抽出を行うことが考えられる。つまり、フレーム毎に特徴点抽出を行い、前フレームの特徴点と対応が付かない現フレームの特徴点を新たに出現した特徴点とみなし追跡対象とする。この方法では、初期フレームだけでなく全てのフレームで特徴点抽出を行うため、計算量が増加する。しかし、特徴点の位置がマッチング以前にわかるため、探索領域におけるマッチングにおいて対応候補を絞ることができる。これにより、探索領域内での枚挙的なマッチングが不要となるため、マッチングに要する計算量は減少する。これらの計算量の増加と減少の関係がフレーム毎に特徴点抽出を行う追跡方法の性質を決定する。その性質を明らかにすることは、その追跡方法の利点と欠点を知り、それを適した用途に使用する上で重要である。

本論文では、探索領域内に存在する特徴点の数の確率分布を用い、フレーム毎に特徴点抽出を行う追跡方法でのマッチング回数の推定を行う。その結果に基づき、追跡に要する計算時間を推定し、その追跡方法の性質を明らかにする。そして、その追跡方

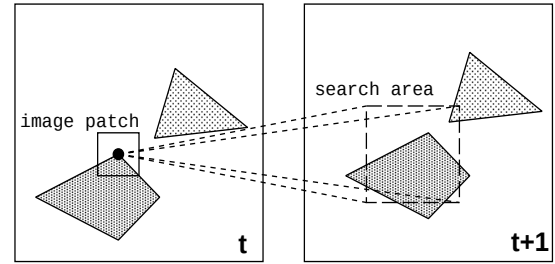


図 1: 探索領域内での枚挙的マッチングによる追跡。

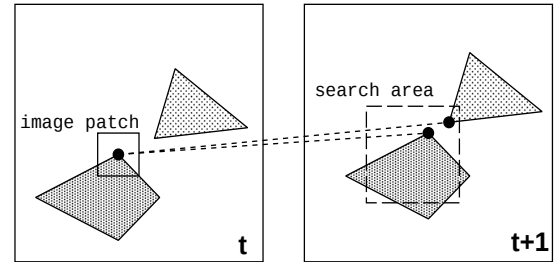


図 2: 特徴点抽出により得られた探索領域内の対応候補とのマッチングによる追跡。

法を実装し、実画像を用いた実験を通じ解析結果を検証する。さらに、時系列画像からのパノラマ画像生成への応用を示す。

2 相関を用いた特徴点の追跡

本論文では、図 1 および図 2 に示す局所領域内での相関マッチングを用いた 2 つの追跡方法を取り扱う。図 1 の方法では、コーナー検出器を用い初期フレームで特徴点抽出を行う。次のフレームで前フレームでの特徴点位置を中心に局所的な探索領域を設定し、その中で前フレームの特徴点位置の近傍の画像データと探索領域内の画像データとの相関を用いたマッチングを行う。そして、相関を最大とする位置を現フレームにおける特徴点の位置とする。これを引き続くフレームでも繰り返し、特徴点の追跡を行う。この方法では、初期フレームのみで特徴点抽出を行うため、視点の移動に伴い視野に出現する特徴点を追跡することができない。

視野に出現する特徴点を追加する方法の一つに、図 2 のようにフレーム毎に特徴点抽出を行う方法がある。つまり、全てのフレームで特徴点抽出を行い、前フレームの特徴点と対応が付かない現フレームの特徴点を新たに出現した特徴点とみなし追跡対象とする。この方法では、特徴点抽出のための計算量は増加する。しかし、その一方で、マッチング以前に特徴点の位置がわかるため、探索領域内での枚

挙的なマッチングを行う方法 (図 1) に比べ、マッチングの計算量は減少する。これらの計算量の増加と減少の関係が、図 1 と図 2 の方法の計算時間の差を決定する。次節では、この両方法の計算時間を推定する。

3 対応付けに要する計算時間の推定

2 フレーム間の対応付けに必要な計算時間を推定する。以下において、 P は特徴点数、 H_i, W_i はピクセルを単位とする画像の高さと幅、 $S_i = H_i W_i$ は画像の面積、 H_s, W_s はピクセルを単位とする探索領域の高さと幅、 $S_s = H_s W_s$ は探索領域の面積、 H_t, W_t はピクセルを単位とするマッチングに用いる画像パッチの高さと幅とする。

3.1 マッチング回数の推定

探索領域内での枚挙的なマッチングを行う場合のマッチング回数は、次式で与えられる。

$$N_e = P S_s \quad (1)$$

フレーム毎に特徴点抽出を行い、探索領域内の特徴点のみに対してマッチングを行う場合のマッチング回数を推定する。探索領域内での枚挙的なマッチングを行う場合と条件を同じにするため、2 フレーム間では特徴点の出現も消失もなく、特徴点数 P は一定と考える。つまり、現フレームでは必ず前フレームと同じ物体上の特徴点が得られるものとする。この場合、マッチング回数は、必ず 1 つある対応点と、探索領域内に存在するそれ以外の特徴点の数により決定される。

マッチング回数を推定するため、探索領域内に存在する対応点以外の特徴点の数の確率分布を求める。画像内に特徴点がランダムに分布すると仮定する。この仮定の下で、各特徴点が探索領域内に存在する確率 p は次式で与えられる。

$$p = \frac{S_s}{S_i} \quad (2)$$

探索領域内には、1 つの対応点と残りの $P-1$ 個の特徴点のうち x 個が存在するとする。 $P-1$ 個の特徴点の中の x 個が同時に探索領域内に存在する確率は、次式で与えられる。

$$\binom{P-1}{x} p^x (1-p)^{(P-1-x)} \quad (3)$$

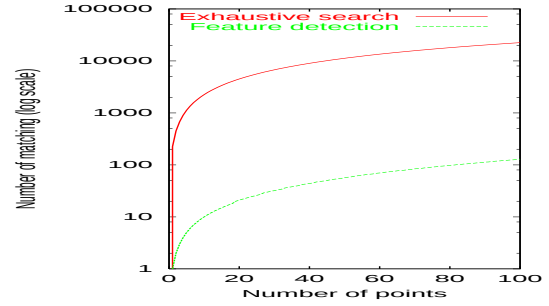


図 3: 特徴点数に対するマッチング回数の変化

つまり、 x は二項分布 $B(P-1, p)$ に従う。この分布の平均は $\mu = (P-1)p$ なので、各特徴点の平均マッチング回数は、

$$1 + \mu = 1 + (P-1)p \quad (4)$$

となる。よって、全特徴点の平均マッチング回数は、次式で与えられる。

$$N_s = P(1 + \mu) = P\{1 + (P-1)p\} \quad (5)$$

一例として $H_i = 240, W_i = 320, H_s = W_s = 15$ とした場合の、マッチング回数 N_e と N_s の特徴点数に対する変化を図 3 に示す。この図から明らかなように、フレーム毎に特徴点抽出を行い対応候補を絞ることによって、マッチング回数が枚挙的な探索を行う場合に比べ大きく減少する。例えば、特徴点数が 50 の場合、 $N_e = 11250$ に対し $N_s = 57$ である。

3.2 計算時間の推定

式 (1) と式 (5) のマッチング回数をを用い、対応付けに必要な計算時間を求める。1 回のマッチングに必要な計算時間を T_m 、1 フレームに対する特徴点抽出に必要な計算時間を T_f とする。枚挙的な探索を行う場合の計算時間は、

$$T_e = P S_s T_m \quad (6)$$

となり、フレーム毎に特徴点抽出を行う場合の平均計算時間は、

$$\begin{aligned} T_s &= P\{1 + (P-1)p\} T_m + T_f \\ &= P^2 p T_m + P(1-p) T_m + T_f \end{aligned} \quad (7)$$

となる。

実際の計算時間を推定するために、次のようにパラメータを設定する: $H_i = 240, W_i = 320; H_t = W_t = 15$ に対し $T_m = 27 [\mu s]$; $H_t = W_t = 21$ に対し $T_m = 48 [\mu s]; T_f = 125 [ms]$ ¹。各画像パッチサ

¹ T_m と T_f の値は、6 節の実験で用いた PC で実現可能な値である。

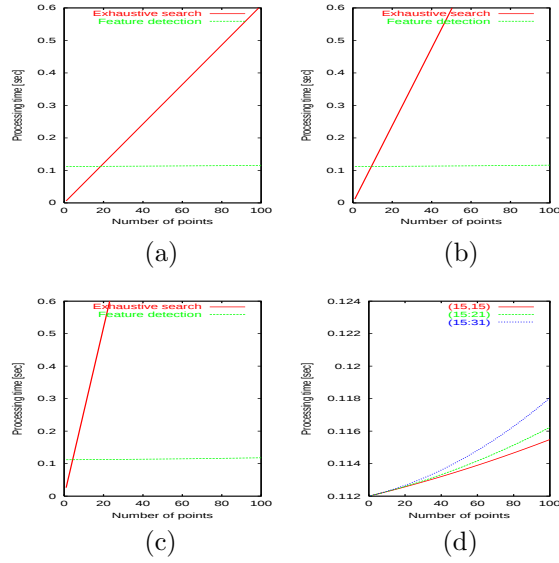


図 4: 特徴点数に対する計算時間の変化: $H_t = W_t = 15$. (a) $H_s = W_s = 15$. (b) $H_s = W_s = 21$. (c) $H_s = W_s = 31$. (d) 特徴点抽出を行った場合の計算時間 (拡大)

イズに対し, $H_s = W_s = 15, 21, 31$ と変化させた場合の計算時間を図 4 と 5 に示す. これらの図から, 次の両方法の性質が明らかとなる.

(1) 探索領域内での枚挙的なマッチングを行う方法は, 特徴点数が少ない場合, 例えば 20 個以下, ならば, 特徴点抽出を行う方法に比べ計算時間が短い (図 4,5(a),(b),(c)).

(2) 特徴点抽出を行う方法は, 特徴点数が多い場合に枚挙的なマッチングを行う方法より計算時間が短い.

(3) 特徴点抽出を行う方法は, 枚挙的なマッチングを行う方法に比べ, 探索領域の大きさ, 特徴点数, 画像パッチの大きさの変化に対する計算時間の変化が非常に小さい (図 4,5(d)).

もちろん, これらに加え,

(4) 特徴点抽出を行う方法は, 視点の移動に伴って視野に出現する特徴点を追跡できる.

という点も両者の差である.

(3) の性質は, 特徴点抽出によりマッチング回数が非常に少なくなるため (図 3), 特徴点抽出に必要な計算時間からの変動が小さくなることによって得られる. これらの性質から, 特徴点抽出を行う方法は, 追跡すべき特徴点が少数に限定できない場合

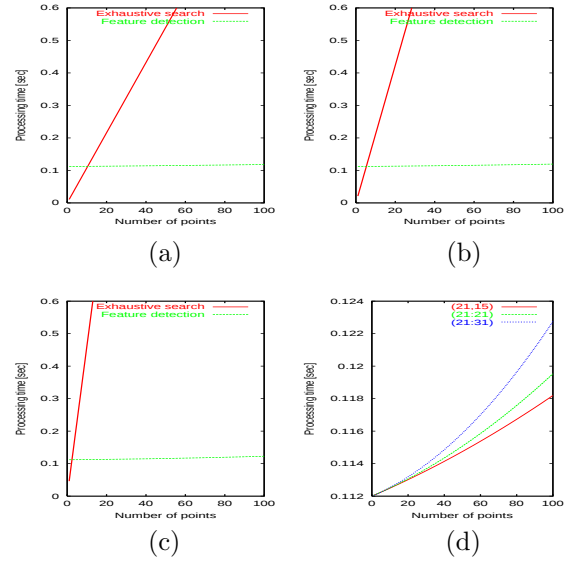


図 5: 特徴点数に対する計算時間の変化: $H_t = W_t = 21$. (a) $H_s = W_s = 15$. (b) $H_s = W_s = 21$. (c) $H_s = W_s = 31$. (d) 特徴点抽出を行った場合の計算時間 (拡大)

や, 計算時間の変動に制限がある (最低のフレームレートに制限がある) 場合に有用と考えられる.

上記の計算時間に関する解析が実際の状況でも妥当であることを, 6 節の実験により示す.

4 特徴点抽出を行う追跡方法の実装

3 節の解析結果を検証するために, 特徴点抽出を行う追跡方法を実装した. 以下にその概要を示す.

4.1 特徴点抽出

特徴点抽出にはコーナー検出器を用いる. 本実装では, Chabat らのコーナー検出器 [11] により得られる画像上の座標 x におけるコーナーらしさ (corneriness) $c(x)$ を用い, 次の手順で特徴点抽出を行った. (1) 各画素に対し $c(x)$ を求める. (2) 15×15 のマスクを用い, $c(x)$ の非極大点抑制処理をする. (3) $c(x)$ のヒストグラムに対する判別分析 [12] により, しきい値 c_{th} を求める. (4) c_{th} より大きな $c(x)$ を持つ座標 x とその数 P を求める. 座標 x は, 非極大点抑制処理前の各画素の $c(x)$ を用いた線形補間により, サブピクセル精度で求める. (5) P が前もって与えた最大特徴点数 P_{max} より大きい場合には, $c(x)$ の大きい順に P_{max} 個を取り出す.

一例として, キャリブレーションパターンに対する特徴点抽出結果を図 6 に示す.

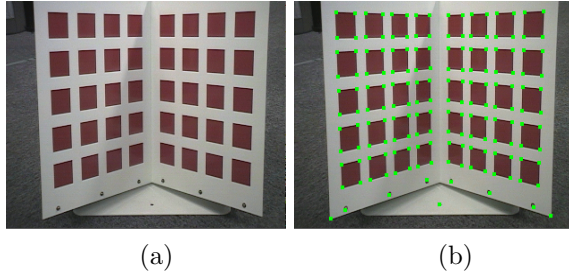


図 6: 特徴点抽出の一例. (a) 原画像. (b) 特徴点抽出結果. 白い四角が特徴点を表す.

4.2 マッチングと競合の解消

現フレームの探索領域内にある特徴点と前フレームの特徴点の近傍の画像パッチを用いて、対応を求める (図 2). マッチングの類似度には、コントラストの変化に対し頑健な正規化相関を用いた.

ある特徴点の対応付けの結果が、その他の特徴点の対応付けの結果と競合する場合がある. その場合、対応付けの結果は一対一対応にならない. 本実装ではこの競合を解消し、一対一対応を得る. 競合解消の手順を図 7 を用いて説明する. 図 7 において、A, B, C は前フレームの特徴点であり、A', B', C' は現フレームの特徴点である. 2 つの特徴点間の正規化相関を、例えば $NC(A, A')$ で表す. 特徴点 A に対し、A' と B' が対応候補となる. この場合、 $NC(A, A')$ と $NC(A, B')$ を比較し、それが大きな方を A の対応点とする. ここで B' が対応点に選ばれたとすると、B' は B の対応点でもあるため競合が生じる. この競合を解消するため、 $NC(A, B')$ と $NC(B, B')$ を比較する. もし、 $NC(A, B') > NC(B, B')$ なら、A が B' と対応すると判断する. もし、 $NC(A, B') < NC(B, B')$ なら、B は B' と対応すると判断し、A は残った A' と対応すると判断する. この場合、A' は C の対応点でもあるため再び競合が生じ、それを解消する必要がある. 全ての競合が解消されるまで、上記の正規化相関の一対比較を繰り返す.

4.3 特徴点の削除と追加

図 7 において、特徴点 A と B', 特徴点 C と A' が対応するとする. この際、特徴点 B は対応点を持たない. この様に、前フレームの特徴点で、現フレームに対応点を持たない特徴点は削除する. また、特徴点 C' は前フレームのいずれの特徴点とも対応しない. このように、前フレームの特徴点と対応しない現フレームの特徴点は、追跡対象として追

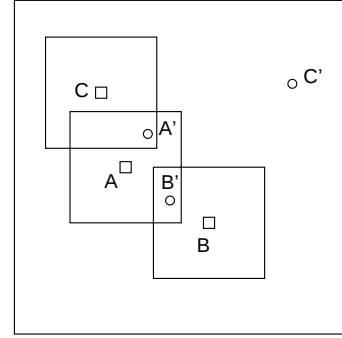


図 7: 特徴点の対応付け. A, B, C は前フレームの特徴点. A', B', C' は現フレームの特徴点.

加する. 追加の結果、特徴点数が P_{max} を越える場合には、 $P_{max} - P_c$ 個 (P_c は現在の特徴点数) の特徴点をコーナーらしさ $c(x)$ の大きい順に選択し追加する.

5 特徴点の信頼度

フレーム毎に特徴点抽出を行うと、連続して抽出される安定した特徴点と、そうでない不安定な特徴点が得られる. これは特徴点抽出により、追跡に適した特徴点のみが抽出されるとは限らないためである. 特徴点抽出の誤りを完全になくすことは非常に困難であるので、安定性の指標を定義し特徴点を区別する. このことは、追跡結果を利用する処理で良い特徴点を選択するためにも重要である. ここでは、特徴点の信頼度を次式で定義する.

$$R_p = \begin{cases} F_c/F_l, & F_c < F_l \\ 1, & F_c \geq F_l \end{cases} \quad (8)$$

ここで F_c は特徴点が現フレームまで連続して追跡されたフレーム数であり、 F_l は前もって与えたフレーム数である. 例えば $F_l = 30$ ならば、30 フレーム以上連続して追跡された特徴点の信頼度は 1 となる.

6 実験

6.1 対応付けに要する計算時間の計測

3 節の解析を検証するために、対応付けに要する計算時間を測定した. 使用した PC の CPU は Pentium Xeon 1.7GHz, OS は Linux(カーネル 2.4.5) である. 画像はカラーで、サイズは 320×240 である.

対応付けを行う 2 フレームの間で特徴点数の変化が生じないように、静止したシーンをを用い、特徴点の位置は初期フレームの位置で固定した. この

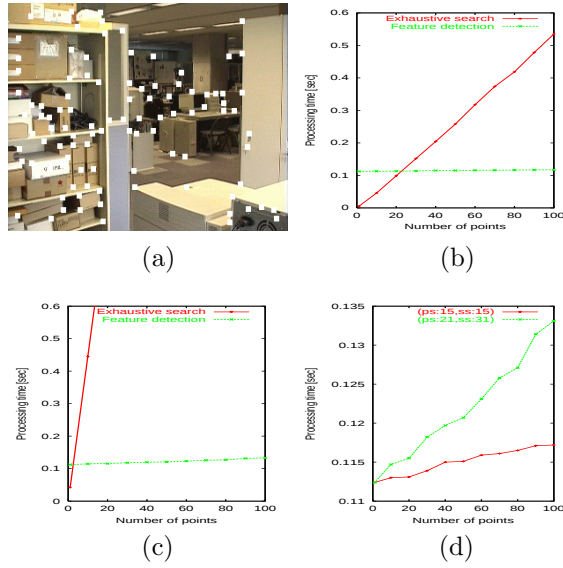


図 8: 対応付けに要する計算時間の計測結果. (a) 計測に使用した画像. 白い四角は特徴点. (b) $H_t = W_t = 15, H_s = W_s = 15$. (c) $H_t = W_t = 21, H_s = W_s = 31$. (d) 特徴点抽出を行った場合の計算時間 (拡大)

場合, 特徴点抽出を行う方法では, 計算時間は探索領域の大きさ, 特徴点数, 画像パッチの大きさと共に, 特徴点の配置に依存する. シーンを図 8(a) に示す. 追跡を行う特徴点数を 1, 10, 20, ..., 100 と変化させ, 対応付けに要した計算時間を計測した. OS の非リアルタイム性による計算時間の変動を考慮し, 300 回同じ画像で対応付けを行い, その平均計算時間を求めた.

枚挙的な探索を行う方法が最も有利になる図 4(a), および, 特徴点抽出を行う方法が最も有利になる図 5(c) の 2 つの条件下で計測を行った. 実際の特徴点の配置は 3 節で仮定したようにランダムでなく偏りがあるが (図 8(a)), 計測結果は 3 節の解析結果と非常に類似している. つまり, この結果より, 枚挙的な探索を行う方法は特徴点数が 20 個程度以下と少ない場合に計算時間が短くなり, 特徴点抽出を行う方法は特徴点が多い場合に計算時間が短くなることがわかる (図 8(b), (c)). また, 特徴点抽出を行う方法では, 探索領域の大きさ, 特徴点数, 画像パッチの大きさの変化に対する計算時間の変化が非常に小さいことがわかる (図 8(d)). この実験では, 特徴点数が 1 から 100 に変化しても, 計算時間の変動は最大で約 21[ms] であった. これらの結果は, 3 節の解析結果が, 実際の追跡方法の性質を良く表

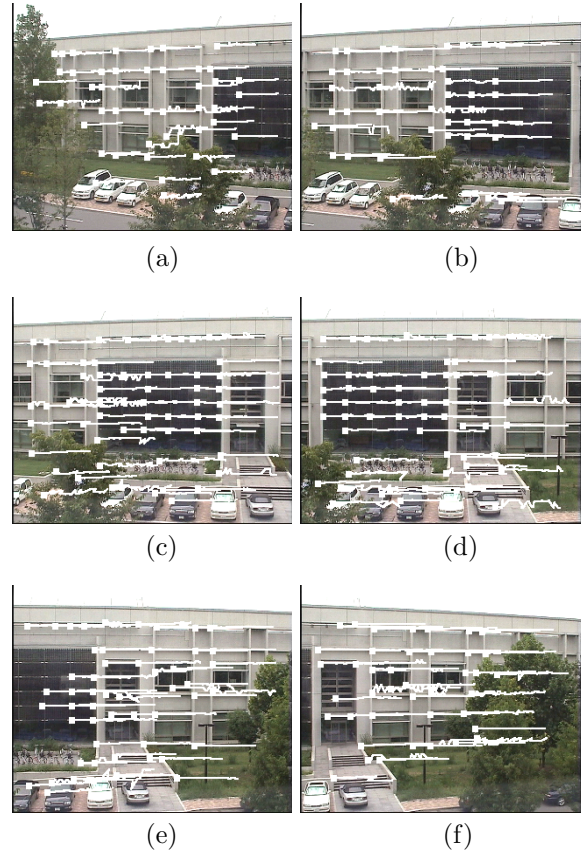


図 9: 追跡結果. (a) Frame 30. (b) Frame 60. (c) Frame 90. (d) Frame 120. (e) Frame 150. (f) Frame 180. 結果を明示するため, $R_p = 1.0 (F_l = 30)$ であった特徴点のみの現フレームでの位置とその前 30 フレームの運動軌跡を示している.

現していることを示している. よって, 3 節の解析結果は, 4 節で示した以外の実装においても, その計算時間の変化の傾向を前もって把握することに使用できると考えられる.

6.2 特徴点の追跡結果

大きな視点の移動を伴う時系列画像において特徴点を追跡した結果を図 9 から図 12 に示す. $H_s = W_s = 31, H_t = W_t = 21, P_{max} = 200$ とした. フレーム毎に特徴点抽出を行うため, 視点の移動に伴い視野に現れる特徴点の追加が行われている (図 9, 11). また, 特徴点数が 70 から 180 程度まで変動した場合の計算時間の変動は, 最大で 40[ms] 程度であった (図 10, 12). この計算時間には, 対応付けに加え, 特徴点の追加と削除に要する計算時間も含まれている. この結果から, 特徴点の出現と消失が生じる場合にも, 特徴点数の変化に対する計算時間

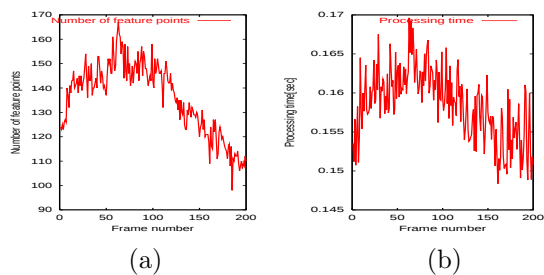


図 10: 図 9 の追跡過程での特徴点数と計算時間の変化。(a) 特徴点数の変化。(b) 計算時間の変化。

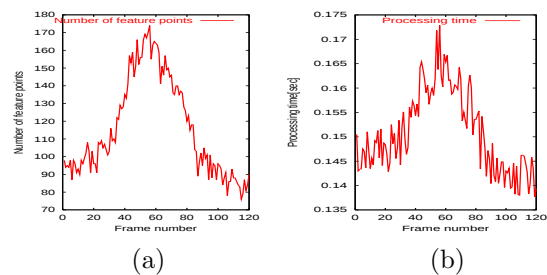


図 12: 図 11 の追跡過程での特徴点数と計算時間の変化。(a) 特徴点数の変化。(b) 計算時間の変化。

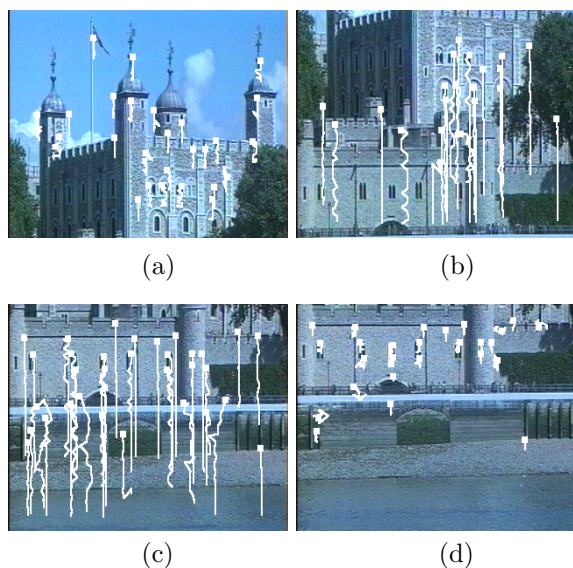


図 11: 追跡結果。(a) Frame 30. (b) Frame 60. (c) Frame 90. (d) Frame 120. 結果を明示するため、 $R_p \geq 0.66 (F_l = 30)$ であった特徴点のみの現フレームでの位置とその前 20 ~ 30 フレームの運動軌跡を示している。

の変化が小さいことが確認された。

6.3 追跡結果を用いたパノラマ画像の生成

フレーム毎に特徴点抽出を行う追跡方法の一応用として、パノラマ画像の生成を行った。追跡結果である特徴点の運動軌跡より、種々のフレーム間隔で対応点が得られる。よって、パノラマ画像を生成する際、合成する画像のフレーム間隔を変化できる。ここでは $R_p = 1.0 (F_l = 30)$ の特徴点の運動軌跡を選び、それらから得られる対応点を用いた。各画像の座標間の変換には、射影変換の部分群であるアフィン変換を用いた。その変換のパラメータ推定の際、誤対応の影響を軽減するため、ロバスト推定の 1 つである RANSAC[13] を用いた。

図 13 に、図 9 の追跡結果より生成したパノラマ



図 14: 図 11 の追跡結果から生成したパノラマ画像。

画像を示す。210 フレーム用い、合成するフレーム間隔を 2 および 30 にした。フレーム間隔が 30 の場合、撮影対象が十分に遠景でないこと等起因すると思われる歪みがパノラマ画像に生じているが、どちらの場合にも追跡結果を用いて画像の合成が可能なのがわかる。図 11 の追跡結果より生成したパノラマ画像を図 14 に示す。120 フレーム用い、フレーム間隔は 30 とした。この場合は撮影対象が遠景であるため、フレーム間隔を粗くしても歪みのないパノラマ画像が得られた。

7 むすび

フレーム毎の特徴点抽出に基づく特徴点の追跡方法について、計算時間に関する解析を行い、その性質について考察した。その結果、探索領域内での枚挙的なマッチングを行う方法に比べ、探索領域の大きさ、特徴点数、マッチングに用いる画像パッチの



図 13: 図 9 の追跡結果から生成したパノラマ画像. 上: 2 フレーム毎に画像を合成した場合. 下: 30 フレーム毎に画像を合成した場合.

大きさの変化に対する計算時間の変化が非常に小さいという性質をもつことが明らかになった. この性質は, 計算時間の変動に制約がある状況で追跡を行う場合に有用である. 実画像へ適用すると共に, パノラマ画像生成への応用を行い, 特徴点抽出を行う追跡方法の有用性を確認した.

参考文献

- [1] I. Zoghlami, O. Faugeras and R. Deriche: "Using geometric corners to build a 2D mosaic from a set of images," Proc. Int. Conf. Computer Vision and Pattern Recognition, pp.420-427, 1997
- [2] C. J. Poelman and T. Kanade: "A paraperspective factorization method for shape and motion recovery," IEEE Trans. Pattern Anal. & Mach. Intell., Vol.19, No.3, pp.206-218, 1997
- [3] N. P. Papanikolopoulos, P. K. Khosla and T. Kanade: "Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision," IEEE trans. Robotics and Automation, vol.9, no.1, pp.14-35, 1993
- [4] N. Ichimura: "Motion segmentation using feature selection and subspace method based on shape space" Proc. Int. Conf. Pattern Recognition, Vol.III, pp.858-864, 2000
- [5] B. D. Lucas and T. Kanade: "An iterative image registration technique with an application to stereo vision," Proc. 7th Int. Joint. Conf. Artificial Intelligence, pp.674-679, 1981
- [6] J. Shi and C. Tomasi: "Good features to track," Int Conf. Computer Vision and Pattern Recognition, pp.593-600, 1994
- [7] A. Goshtasby, S. H. Gage, and J. F. Bartholic: "A two-stage cross correlation approach to template matching," IEEE trans. Pattern Anal. & Mach. Intell., vol.6, no.3, pp.374-378, 1984
- [8] M. W. Eklund, G. Ravichandran, M. M. Trivedi and S. B. Marapane: "Real-time visual tracking using correlation technique," Proc. 2nd IEEE Workshop on Application of Computer Vision, pp.256-263, 1994
- [9] Q. Zheng and R. Chellappa: "Automatic feature point extraction and tracking in image sequences for arbitrary camera motion," Internat. J. Comp. Vis., vol.15, pp.31-76, 1995
- [10] J. P. Lewis: "Fast template matching," Vision Interface, pp.120-123, 1995
- [11] F. Chabat, G.Z. Yang and D.M. Hansell: "A corner orientation detector," Im. and Vis. Comp., 17, pp.761-769, 1999
- [12] N. Otsu: "A threshold selection method from gray-level histograms," IEEE Trans. Sys., Man, and Cybern., Vol.SMC-9, No.1, pp.62-66, 1979
- [13] M. A. Fischler and R. C. Bolles: "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," Commun. of the ACM, Vol.24, No.6, pp.381-395, 1981