

シーン拘束を用いた人間動作の高精度動画像追跡

Precise Tracking of Human Motion by Scene and Image Constraints

八木下 勝利†, 正会員 山本 正信†

Katsutoshi Yagishita† and Masanobu Yamamoto†

Abstract This paper describes a new method for tracking objects in 3D motion from image sequences by using constraints in the scene. Many methods based on the gradient scheme have been proposed for tracking objects. However, these methods have a serious drawback in that tracking error sometimes increases during a long image sequence due to various noise. The proposed method enables us to track the objects over a long time by the gradient-based constraint with some scene constraints; that is, constraints on position, velocity, and acceleration in the scene. Experiments with real image sequences support the precise tracking over a long time.

キーワード：シーン拘束，動画像追跡，時空間勾配法，多関節ロボットモデル

1. ま え が き

人間の動作を定量的に捉えることができれば，計算機と人間のインタフェース，CGアニメーションなどにおけるキャラクターの動作の生成，防犯のためのモニタリング，計算機による行動理解への応用など，数多くの分野での活用が期待される。

人体のような複雑な対象の動きを測定する方法のひとつとして，モーションキャプチャ等の特殊な装置の利用が挙げられる。しかし，この方法は装置が大がかりとなる上に，対象に特殊なデバイスを取り付けることが多い。そのため対象の運動が制限されるといった問題がある。

一方，普通のビデオカメラから得られた画像を使っても動作を測定することができる。この場合は特殊なデバイスは不要であり，対象の自由で自然な動作を測定することが可能である。さらに，映像がある限り過去のものであってもその運動の取得が可能である。実際，筆者らのグループにより提案された動画像解析法¹⁾では，時空間勾配法とロボットモデルを用いることで人体の動作を3次元的に測定することが可能となった。しかしながら，測定が長時間となると運動を正確に測定するのが難しくなる。これは，種々の雑音の影響によって生じる誤差が累積されることにより，追跡対象の実際の位置・姿勢と求められた位置・姿

勢との間にずれが生じるためである。

そこで本論文では，時空間勾配法とロボットモデルに基づく動画像追跡において，シーンから対象に働く拘束を考慮することを提案する。これにより，実際の位置・姿勢と求められた位置・姿勢との間に生じるずれを抑え，長時間，精度の高い追跡を行うことができる。シーンからの拘束を利用するアプローチとしては Leung らの手法²⁾があるが，これは拘束の適用が不動部分の位置に限定されており，人体のモデルも2次元である。これに対して，本手法では人体モデルとして3次元モデルを用いているため，3次元での運動が推定できる。また，シーンからの拘束は運動している点に対して適用でき，その点の位置，速度，加速度を拘束することが可能である。

本論文では，まず2章で追跡対象の運動をモデル化し，3章で時空間勾配拘束について述べる。そして，4章においてシーン拘束による拘束式を示し，それらを5章において時空間勾配拘束と融合する。6章では本手法を用いて追跡実験を行い，その有効性を示す。

2. 運動モデル

追跡対象を図1のように多関節ロボットモデルで表す。これは，対象を関節で分解し，その各セグメントごとにソリッドモデルとして作成したものである³⁾。このとき，ロボットモデルの全関節の位置と姿勢を表すベクトル $\mathbf{q} = (q_1, q_2, \dots, q_n)^T$ を用いて，モデル(追跡対象)上の点の位置 $\mathbf{p} = (x, y, z)^T$ を，以下のように表すことができる⁴⁾。

$$\mathbf{p} = \mathbf{f}(\mathbf{q}) \quad (1)$$

1997年3月12日，情報処理学会全国大会で一部発表
1997年8月29日受付，1997年11月25日再受付，1997年12月17日採録
†新潟大学工学部情報工学科

(〒950-21 新潟市五十嵐2の町 8050, TEL 025-262-7462)

†Department of Information Engineering, Niigata University
(8050, Ikarashi 2-nocho, Niigata 950-21, Japan)

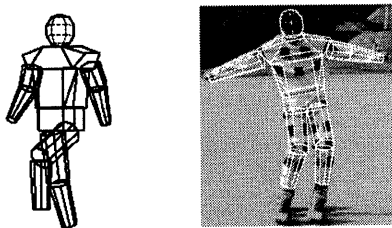


図1 ロボットモデルと画像との照合
Fitting a robot arm model to an image.

ここで f は一般には非線形関数であるが、 q の時間変化率 \dot{q} と点 p での速度ベクトル \dot{p} は、ロボットモデルの位置と姿勢により決定されるヤコビ行列 $J(q)$ を用いて次の線形式で関係付けられる。

$$\dot{p} = J(q)\dot{q} \quad (2)$$

ここで \dot{q} はロボットモデルの動きを表すもので、運動パラメータと呼ぶ。また、実際のヤコビ行列の計算法は文献1)を参照されたい。

3. 時空間勾配拘束

動画像の濃淡値を $E(X, Y, t)$ と表す。ここで、 (X, Y) は画像上の点の座標値、 t は時刻とする。このとき、画像上での各点の速度ベクトル (U, V) は、次の時空間勾配式に従う³⁾。

$$E_X U + E_Y V + E_t = 0 \quad (3)$$

ここで E_X, E_Y はそれぞれ E の X 方向 Y 方向の変化率(空間勾配)であり、 E_t は E の時間変化率(時間勾配)である。(3)式を3次元空間の運動へ拡張する。

カメラは中心投影モデルとし、焦点距離を f とする。シーンの座標系は、その原点を視点とし、 z 軸を光軸の向きにとり、 x, y 軸は投影面の X, Y 軸と平行となるように設定する。このとき、3次元シーン中の点 $p = (x, y, z)^T$ は、投影面上の点 (X, Y) に次の変換式で投影される。

$$\begin{cases} X = f \frac{x}{z} \\ Y = f \frac{y}{z} \end{cases} \quad (4)$$

また、3次元速度ベクトル $\dot{p} = (u, v, w)^T$ と投影面上での速度ベクトル (U, V) は、(4)式を時刻 t で微分することで、以下のように得られる。

$$\begin{cases} U = \frac{fu - Xw}{z} \\ V = \frac{fv - Yw}{z} \end{cases} \quad (5)$$

(5)式を(3)式に代入すれば

$$G^T \dot{p} = -E_t \quad (6)$$

となる。ここで

$$G = \left(f \frac{E_X}{z}, f \frac{E_Y}{z}, -\frac{X E_X + Y E_Y}{z} \right)^T \quad (7)$$

である。さらに(6)式から、(2)式により \dot{p} を消去すれば

$$G^T J \dot{q} = -E_t \quad (8)$$

となる。この式が、動画像の時空間勾配による運動パラメータ \dot{q} の拘束式となる。

時空間勾配法とロボットモデルに基づく動画追跡では、まず図1のようにロボットモデルを作成し初期フレーム上で実際の対象と合致させておく。人体などは非剛体であり、また正確な大きさがわからないため、単眼視では対象の位置・姿勢が正確には求められないが、投影面上での実際の対象の像とモデルの投影像とが一致していれば、3次元シーン空間においてもほぼ一致しているものとする。その後、1フレーム毎にロボットモデル上(人体像上)の複数個の点に対しそれぞれ運動パラメータの推定式(8)を求め、それらを次式のように連立させる。

$$\begin{pmatrix} G_1^T J_1 \\ G_2^T J_2 \\ \vdots \\ G_n^T J_n \end{pmatrix} \dot{q} = \begin{pmatrix} -E_{t1} \\ -E_{t2} \\ \vdots \\ -E_{tn} \end{pmatrix} \quad (9)$$

ここで添字は各点を表している。この連立方程式(9)を最小二乗法を用いて解くことで運動パラメータ \dot{q} を推定し、その推定値に基づきロボットモデルを移動させることにより追跡を行う。

4. シーン拘束

動画追跡の際に、追跡対象の動きは、シーン中の他のものから拘束されている場合が多い。例えば、歩く動作では足の位置は地面に拘束され、ドアを開ける動作では手の位置がドアの取手の運動軌跡に従い、列車の動きはその軌道に拘束される。また、スキーなど面上を滑る運動ではその速度の向きが滑る面と平行に拘束されるし、重力による落下運動では加速度の向き(追跡対象にはたらく力の向き)が拘束される。このような、シーン中から追跡対象上のある点の位置、速度、および加速度に働く拘束をシーン拘束と呼ぶことにし、これを3章で述べた時空間勾配拘束に加え、動画追跡を行う。

4.1 位置の拘束

あるフレームにおいて、追跡対象上の任意の点 p の次フレームへの運動が、法線ベクトル n 、原点からの有向距離 d である平面

$$n^T x = d \quad (10)$$

に拘束されているものとする。 p の次フレームでの位置 p' は、増分を Δp として

$$p' = p + \Delta p \quad (11)$$

となるが、この点 p' も (10) 式で表される平面上にあるので

$$\mathbf{n}^\top(\mathbf{p} + \Delta\mathbf{p}) = d \quad (12)$$

が成立する。ここで、フレーム間隔を単位時間とし、またフレーム間の移動距離は充分小さいものとして

$$\Delta\mathbf{p} = \dot{\mathbf{p}} \quad (13)$$

とおく。(12) 式を (13) 式を用いて書き換えて

$$\mathbf{n}^\top(\mathbf{p} + \dot{\mathbf{p}}) = d \quad (14)$$

となる。さらに (14) 式から、(2) 式により $\dot{\mathbf{p}}$ を $\dot{\mathbf{q}}$ に変換して次式を得る。

$$\mathbf{n}^\top \mathbf{J} \dot{\mathbf{q}} = d - \mathbf{n}^\top \mathbf{p} \quad (15)$$

これが追跡対象上の点の位置を平面上に拘束するための、運動パラメータ $\dot{\mathbf{q}}$ の拘束式となる。

点の位置を曲面上に拘束する場合は、フレーム間の動きは充分に小さいものとして、各フレームでの拘束面をそのフレームでの被拘束点に最も近い接平面で近似し、その接平面で拘束を行う。これは、(15) 式において、平面の法線ベクトル \mathbf{n} が点の位置 \mathbf{p} の関数になったと考えればよい。また、点の位置を曲線や固定点上に拘束する場合は、2 曲面の交わりが曲線、3 平面の交わりが 1 点となることを利用し、曲線、固定点を、それぞれ 2 つの曲面、3 つの平面に分解して、それらで拘束を行う。

拘束を与える平面の取得方法は、形状が既知の物体が投影されていれば、透視変換の逆変換によって拘束面を求めることができる⁸⁾⁹⁾。そうでない場合には、3 章で述べた初期フレームでのロボットモデルと実際の対象との照合を複数フレームにおいて行い、その結果での被拘束点の位置から拘束面を求める。

4.2 速度の向きの拘束

追跡対象上の点 \mathbf{p} の速度 $\dot{\mathbf{p}}$ の方向が、ある方向 $\mathbf{s} = (s_x, s_y, s_z)^\top$ に拘束されているとする。これは、 $\dot{\mathbf{p}}$ と \mathbf{s} が同じ方向ということなので

$$\mathbf{s} \times \dot{\mathbf{p}} = \mathbf{0} \quad (16)$$

が成立する。(2) 式を用いて、(16) 式の $\dot{\mathbf{p}}$ を $\dot{\mathbf{q}}$ で書き換えると、以下ようになる。

$$\mathbf{s} \times (\mathbf{J} \dot{\mathbf{q}}) = \mathbf{0} \quad (17)$$

ここで (17) 式中の外積を展開すると

$$\begin{cases} (0, -s_z, s_y) \mathbf{J} \dot{\mathbf{q}} = 0 \\ (s_z, 0, -s_x) \mathbf{J} \dot{\mathbf{q}} = 0 \\ (-s_y, s_x, 0) \mathbf{J} \dot{\mathbf{q}} = 0 \end{cases} \quad (18)$$

となる。これが点の速度の向きをあるベクトル \mathbf{s} と平行に拘束するための、運動パラメータ $\dot{\mathbf{q}}$ の拘束式となる。

次に、ある点 \mathbf{p} の速度 $\dot{\mathbf{p}}$ の方向が、法線ベクトル \mathbf{n} の平面と平行に拘束されているとすれば

$$\mathbf{n}^\top \dot{\mathbf{p}} = 0 \quad (19)$$

が成立する。(2) 式を用い、(19) 式を書き換えて

$$\mathbf{n}^\top \mathbf{J} \dot{\mathbf{q}} = 0 \quad (20)$$

となる。これが点の速度の方向を法線ベクトル \mathbf{n} の平面と平行に拘束するための、運動パラメータ $\dot{\mathbf{q}}$ の拘束式である。

4.3 加速度の向きの拘束

対象物体上の点 \mathbf{p} の加速度 $\ddot{\mathbf{p}}$ の向きが、あるベクトル $\mathbf{s} = (s_x, s_y, s_z)^\top$ の方向に拘束されていれば、速度の向きの拘束の場合と同様に

$$\mathbf{s} \times \ddot{\mathbf{p}} = \mathbf{0} \quad (21)$$

が成立する。ここで加速度の変化は充分に小さいと考えられ、またフレーム間隔を単位時間としたので、加速度 $\ddot{\mathbf{p}}$ は速度 $\dot{\mathbf{p}}$ の差分として次のように書き換えられる。

$$\ddot{\mathbf{p}}_i = \dot{\mathbf{p}}_i - \dot{\mathbf{p}}_{i-1} \quad (22)$$

(22) 式を (21) 式に代入すれば

$$\mathbf{s} \times \dot{\mathbf{p}}_i = \mathbf{s} \times \dot{\mathbf{p}}_{i-1} \quad (23)$$

となる。さらに、(2) 式を用いて (23) 式を書き換えるが、ここで

$$\begin{aligned} \Delta\mathbf{p}_{i-1} &= -(\mathbf{J}_i(-\Delta\mathbf{q}_{i-1})) \\ &= \mathbf{J}_i \Delta\mathbf{q}_{i-1} \approx \mathbf{J}_i \dot{\mathbf{q}}_{i-1} \end{aligned} \quad (24)$$

より、ヤコビ行列は現フレーム i での値だけを使うこととして

$$\mathbf{s} \times \mathbf{J} \dot{\mathbf{q}}_i = \mathbf{s} \times \mathbf{J} \dot{\mathbf{q}}_{i-1} \quad (25)$$

を得る。そして、(25) 式中の外積を展開することで、以下のようになる。

$$\begin{cases} (0, -s_z, s_y) \mathbf{J} \dot{\mathbf{q}}_i = (0, -s_z, s_y) \mathbf{J} \dot{\mathbf{q}}_{i-1} \\ (s_z, 0, -s_x) \mathbf{J} \dot{\mathbf{q}}_i = (s_z, 0, -s_x) \mathbf{J} \dot{\mathbf{q}}_{i-1} \\ (-s_y, s_x, 0) \mathbf{J} \dot{\mathbf{q}}_i = (-s_y, s_x, 0) \mathbf{J} \dot{\mathbf{q}}_{i-1} \end{cases} \quad (26)$$

これが点の加速度の向きをあるベクトル \mathbf{s} と平行に拘束するための、運動パラメータ $\dot{\mathbf{q}}$ の拘束式となる。

また、加速度 $\ddot{\mathbf{p}}$ の向きが法線ベクトル \mathbf{n} の平面と平行に拘束されている場合には

$$\mathbf{n}^\top \ddot{\mathbf{p}} = 0 \quad (27)$$

となる。ここでも (22), (2), (24) 式を用いて書き換えることで、(27) 式は、

$$\mathbf{n}^\top \mathbf{J} \dot{\mathbf{q}}_i = \mathbf{n}^\top \mathbf{J} \dot{\mathbf{q}}_{i-1} \quad (28)$$

となる。これが点の加速度の向きを法線ベクトル \mathbf{n} の平面と平行に拘束するための、運動パラメータ $\dot{\mathbf{q}}$ の拘束式となる。

速度、加速度の拘束の場合にも曲面上に拘束する場合には、曲面をその接平面で近似する。

5. 時空間勾配拘束とシーン拘束の融合

3章の(8)式, 4章の(15), (18), (20), (26), (28)式により動画像中の物体に働く様々な拘束が表された. それらはすべて, 運動パラメータ $\dot{\mathbf{q}}$ に関する1次方程式で表してあるので, それらを連立させることで容易に融合することができ, 以下の式を得る.

$$\begin{pmatrix} w_1 \mathbf{G}_1^\top \mathbf{J}_1 \\ \vdots \\ w_j \mathbf{n}_j^\top \mathbf{J}_j \\ \vdots \\ w_k \hat{\mathbf{s}}_{kx}^\top \mathbf{J}_k \\ w_k \hat{\mathbf{s}}_{ky}^\top \mathbf{J}_k \\ w_k \hat{\mathbf{s}}_{kz}^\top \mathbf{J}_k \\ \vdots \\ w_l \mathbf{n}_l^\top \mathbf{J}_l \\ \vdots \\ w_m \hat{\mathbf{s}}_{mx}^\top \mathbf{J}_m \\ w_m \hat{\mathbf{s}}_{my}^\top \mathbf{J}_m \\ w_m \hat{\mathbf{s}}_{mz}^\top \mathbf{J}_m \\ \vdots \\ w_n \mathbf{n}_n^\top \mathbf{J}_n \\ \vdots \end{pmatrix} \dot{\mathbf{q}}_i = \begin{pmatrix} -w_1 E_{t1} \\ \vdots \\ w_j (d_j - \mathbf{n}_j^\top \mathbf{p}_j) \\ \vdots \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ w_m \hat{\mathbf{s}}_{mx}^\top \mathbf{J}_m \dot{\mathbf{q}}_{i-1} \\ w_m \hat{\mathbf{s}}_{my}^\top \mathbf{J}_m \dot{\mathbf{q}}_{i-1} \\ w_m \hat{\mathbf{s}}_{mz}^\top \mathbf{J}_m \dot{\mathbf{q}}_{i-1} \\ \vdots \\ w_n \mathbf{n}_n^\top \mathbf{J}_n \dot{\mathbf{q}}_{i-1} \\ \vdots \end{pmatrix} \quad (29)$$

ただし

$$\begin{cases} \hat{\mathbf{s}}_{ox} = (0, -s_{oz}, s_{oy})^\top \\ \hat{\mathbf{s}}_{oy} = (s_{oz}, 0, -s_{ox})^\top \\ \hat{\mathbf{s}}_{oz} = (-s_{oy}, s_{ox}, 0)^\top \end{cases} \quad (30)$$

とする. ここで(29)式の係数行列, 定数ベクトルを, 説明の便宜上, 式中のように6つのブロックに区分けする. 第一ブロックは時空間勾配拘束による拘束式(8)で, 各文字の添字は時空間勾配拘束式を求めた点を表す. 第二ブロック以下は, シーン拘束による拘束式で, 順に, 位置の拘束による拘束式(15), 速度の拘束による拘束式(18), (20), 加速度の拘束による拘束式(26), (28)である. 各文字の添字は拘束面, 被拘束点の組合せを表す. 各拘束が曲面によって与えられている場合には, 各拘束式における平面のパラメータ \mathbf{n} , d が, 時刻 t すなわちフレーム番号 i の関数となる. また各拘束が曲線や点によって与えられている場合には, 1つの被拘束点から複数の拘束式が求まることになる. この連立方程式(29)を最小二乗法を用いて解き, 運動パラメータ $\dot{\mathbf{q}}$ を推定することで, 時空間勾配拘束とシーン拘束の両者を考慮した動画像追跡が可能となる.

(29)式の w は, 各拘束をどの程度重視するかという重みである. 各シーン拘束による拘束式を満たすような動きを表す運動パラメータは無数に存在する. そのため, シーン拘束の重みを大きくしすぎると(29)式が一意な解を持た

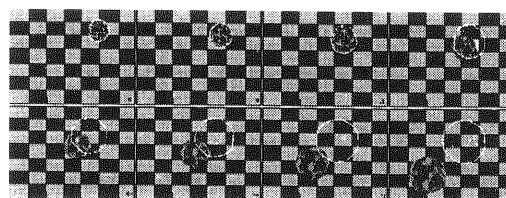


図2 時空間勾配法のみでの追跡
Tracking result by the only gradient-based method.

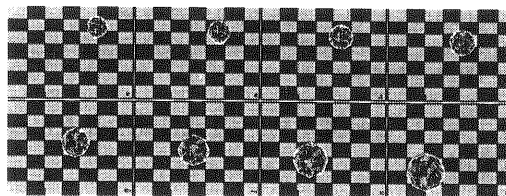


図3 シーン拘束を加えての追跡
Tracking result by the gradient-based method with scene constraints.

ないものとなるので, 慎重に決定する必要がある.

6. 追跡実験

本手法を用いて行った追跡の結果を示す. 6.1節ではシミュレーション画像の追跡を行うことで本手法の効果を定量的に示し, 6.2節, 6.3節では実画像においての有効性を確認する.

6.1 シミュレーション画像の追跡

ボールが斜面を転がるという70フレームのシミュレーション画像を作成し, 追跡を行った. 作成した動画像には正規性雑音を付加した. シーン拘束は, 転がっている平面により与えられる位置の拘束, および速度・加速度の向きの拘束を用いた. 速度・加速度の拘束は, 重心の速度および加速度を転がっている斜面に平行に拘束する. 図2, 3に平均0, 分散10の正規雑音を付加した場合の追跡結果を示す. この画像では背景に強いテクスチャエッジがあるため, 時空間拘束のみでの追跡では, 雑音によりモデルと実際の追跡対象がずれるとその先の追跡がいちじるしく困難となっているが, シーン拘束を用いることでずれが抑えられ追跡が可能となっている. このとき最終フレームにおいて, 並進成分の全体の移動距離に対する相対誤差は2.7%, 実際の姿勢行列と求められた姿勢行列の差のノルムは0.12であった.

また, シーン拘束の重みを変化させたときの追跡誤差を図4に示す. ここで, 時空間勾配拘束の重みはすべて1とし, シーン拘束の重みはすべて同じ値で変化させた. さらに, この場合は背景を一樣な明るさとした. シーン拘束の重みを大きくしすぎると, 運動パラメータの推定式が不安定になり, 正確な追跡が行えなくなっている. シミュレーション実験では, 重みを4とした.

6.2 扉を開閉する腕の追跡

扉を開閉する動作の追跡を行った. 動画像は100フレ

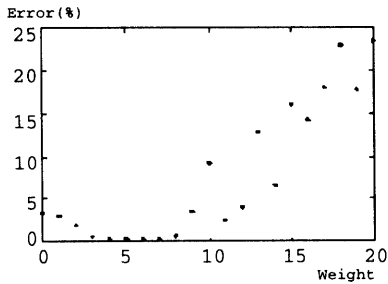


図 4 シーン拘束の重みと追跡誤差

Tracking error versus the magnitude of weight of scene constraints.

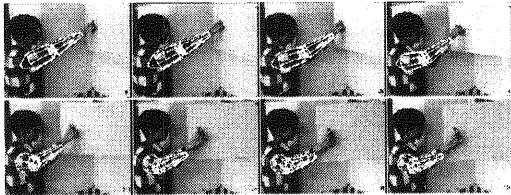


図 5 時空間勾配法のみでの追跡

Tracking result by the only gradient-based method.

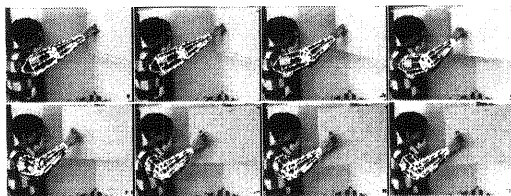


図 6 シーン拘束を加えての追跡

Tracking result by the gradient-based method with scene constraints.

ムである。この場合のシーン拘束は位置の拘束を用い、扉の回転軸を中心軸とする円筒面と扉の取手の高さを示す平面によって与えられる曲線上に手先を拘束する。2つの拘束面は実際に測定した扉の形状から求めた。図5, 6に追跡結果を示す。この場合には2つの面から手先の軌跡が完全に与えられるので、手先においてのずれはほとんどなくなり、その結果として腕全体での追跡もほぼ正確に行われている。

6.3 グレステンスキーの動作の追跡

グレステンスキー*の動作の追跡を行った。動画は150フレームである。この場合のシーン拘束は位置の拘束および速度の拘束を用いた。位置の拘束は両足先をゲレンデの斜面上に拘束し、速度の拘束は両足先および人体の重心の速度を斜面と平行に拘束する。これらの拘束面は、先頭フレーム、中間フレーム、および最終フレームにおいて、実寸で作成されたモデルと人体像を照合した結果から求めた。図7, 8に追跡結果を示す。図9からもわかるように、時空間勾配法のみでの追跡では、追跡後半において両足の姿

* ホンマ科学株式会社(新潟県十日町市)により開発された、特殊なタイヤで陸上を滑走するスキー。雪上でのスキーに近い感覚が得られるため、アルペンスキーのトレーニングなどにも利用されている。

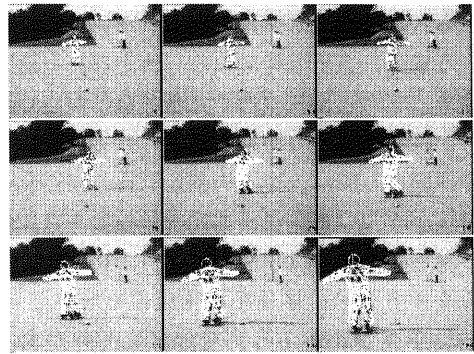


図 7 時空間勾配法のみでの追跡

Tracking result by the only gradient-based method.

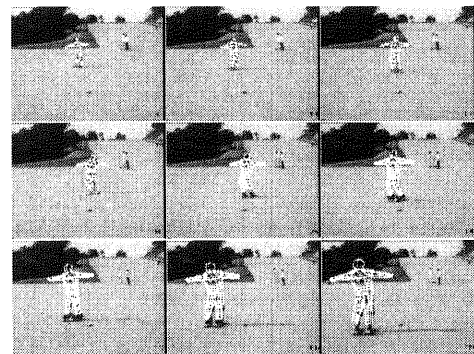


図 8 シーン拘束を加えての追跡

Tracking result by the gradient-based method with scene constraints.

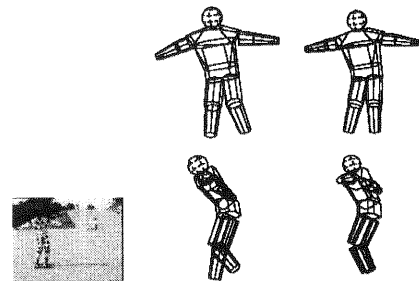


図 9 最終フレームとそこでの姿勢(左:時空間勾配法のみ, 右:シーン拘束を加えたもの)

Left is the final frame. Center: 3D pose at the final frame by the only gradient-based method. Right: 3D pose at the final frame by the gradient-based method with scene constraints.

勢が完全にくずれてしまっている。一方、シーン拘束を用いた追跡では、拘束面(ゲレンデの斜面)と平行な方向に対してのずれは生じるものの、両足の姿勢は実際の人体の姿勢に近くなっている。

7. む す び

動画追跡にシーン拘束を考慮することを提案し、実際に時空間勾配法とロボットモデルに基づく動画追跡に導入した。またこの手法を用いた追跡実験を行い、その効果を確認した。追跡結果からもわかるように、シーン拘束は動画追跡の精度向上に有効な手段であるといえる。本

手法は、あらかじめカメラに対するシーンの構造が容易に測定できる場合に適用可能である。例えば、室内の監視装置の映像や、フィールドの形や大きさなどが定められているスポーツ映像などへの適用が考えられる。

今後の課題としては、まず拘束の取得の自動化が挙げられる。形状が既知の物体が動画像中に投影されていない場合には、現在のところは複数のフレームにおいてモデルと追跡対象を照合しその結果から拘束面を取得しているが、この方法は手間がかかるうえ精度にも問題がある。また(29)式の重みが追跡結果に大きく影響するため、その最適化も必要である。

〔文 献〕

- 1) 山本正信, 川田聡, 近藤拓也, 越川和忠: “ロボットモデルに基づく人間動作の3次元動画像追跡”, 信学論, **J79-D-II**, 1, pp.71-83 (1996)
- 2) M.K.Leung, Y.H.Yang: “First Sight: A Human Body Outline Labeling System”, IEEE Trans.PAMI, **17**, 4, pp.359-377 (1995)
- 3) B.K.P.Horn and B.G.Schunck: “Determining Optical Flow”, Artificial Intelligence, **17**, 1-3, pp.185-203 (1981)
- 4) K.Koshikawa and Y.Shirai: “A 3-D Modeler Vision Research”, Proc. of the Int. Conference on Advanced Robotics, pp.185-190 (1985)
- 5) 倉掛正治, ラマカント ネパティヤ: “関節のある動物体の記述と追跡”, 信学論, **J76-D-II**, 4, pp.854-862 (1993)
- 6) K.Rohr: “Towards Model-based Recognition of Human Movements in Image Sequences”, CVGIP: Image Understanding, **59**, 1, pp.94-115 (1994)
- 7) R.P.Paul: “Robot Manipulators, Mathematics, Programming, and Control”, The MIT Press, Cambridge Massachusetts (1981)
- 8) D.DeMenthon, L.S.Davis: “Exact and Approximate Solutions of the Perspective-three-point Problem”, IEEE Trans.PAMI, **14**, 11, pp.1100-1108 (1992)
- 9) D.G.Lowe: “Fitting Parameterized Three-dimensional Models to Images”, IEEE Trans.PAMI, **13**, 5, pp.441-450 (1991)



やぎしたかつとし
八木下勝利 1997年, 新潟大学工学部情報工学科卒業。現在, 同大学院自然科学研究科博士前期課程在学中。1997年, 情処学会奨励賞受賞。コンピュータビジョンに関する研究に従事。



やまもとまさのぶ
山本 正信 1973年, 九州工業大学工学部制御工学科卒業。1975年, 東京工業大学大学院修士課程修了。同年, 電子技術総合研究所に入所。コンピュータビジョンの研究に従事。1989-1990年, カナダ国立研究協議会招聘研究員。1987年, 情処学会研究賞受賞。1992年より, 新潟大学工学部情報工学科教授。工学博士。正会員。