Vol.2017-CVIM-205 No.6
2017/1/19

3次元特徴量のスパース表現を用いた点群リアルタイムトラッキング

宇山 慧佑[1,a)]　井上 真郷[1,b)]

概要：小型化・低価格化が進む RGB-D センサでの主要な処理に物体のトラッキングが挙げられる．本研究では，特定視点の 1 フレーム分のみのテンプレートを使い，点群データ中の物体をリアルタイムにトラッキングする手法を開発した．点群に過剰に領域分割を施すことで探索を限定，高速化し，映像における L1 最適化によるトラッキングの枠組みを 3 次元空間でのリアルタイムトラッキングに拡張できるようにした．物体の完全な情報をもたないテンプレートでありながら，3 次元特徴量をスパース表現する基底を抽出することで，追跡物体の姿勢変化，オクルージョン，変形にロバストなトラッキングを実現した．

1. はじめに

本論文では，RGB-D センサを用いて，1 フレームの深度画像のテンプレートを使い，リアルタイムに物体をトラッキングする手法について説明する．本手法は，物体のモデルを必要とせず，1 フレーム分の特定視点のテンプレートのみを使うことで，わずかな学習時間で物体の大きな姿勢変化や変形，オクルージョンにロバストなトラッキングを可能にした（**図 1**，**図 2**）．

近年急速に小型化や低価格化の進む，RGB-D センサによるトラッキングは，ロボットのマニピュレーションや拡張現実デバイス，ヒューマンマシンインターフェースなどさまざまな応用の基礎となる技術の 1 つである．既知の完全なモデルに対してトラッキングする手法は多く研究されているが，実際の応用では，人間のように未知の物体や変形を伴う物体に対しても物体を捉えることができる必要がある．本手法では，辞書学習による柔軟性のある特徴を使うことで，1 フレーム分のテンプレートであっても本来学習していない物体の裏側や大きな変形のトラッキングに対応できる．また，あらかじめ表面特徴や空間的な特徴を用いてフレーム毎に小領域に領域分割を行うことで、正確なトラッキングとリアルタイム処理を両立した．

関連研究

トラッキングする物体について 3 次元モデルがわかっている場合，事前に得られた様々な角度からの物体の見え方を知見として利用し高速なトラッキングが行われている [6]．さらに高速でロバストな処理を可能にするため，ランダムフォレストによりこのような様々な見え方の深度データや合成によって生成されたデータを学習しトラッキングを行う手法が多く研究されている [1][5][9][10][13][14][11]．特に文献 [10] では，リアルタイムにランダムフォレストの木を学習することでモデルフリーのトラッキングを実現している．しかし，ランダムフォレストを用いた手法は高速化が可能であるが，ロバストなトラッキングを実現するには様々な状態のデータを用意する必要であるという欠点があ

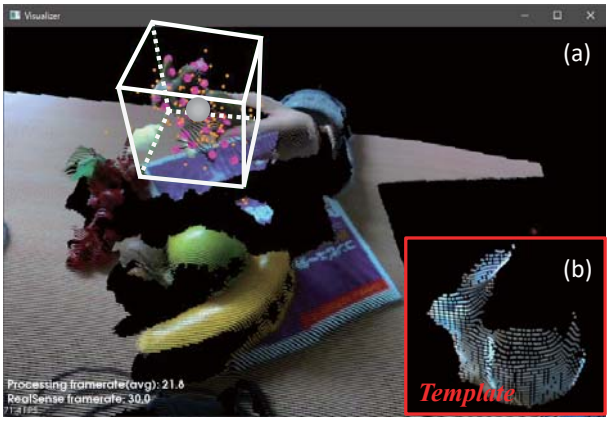

図 1　提案手法による物体のトラッキングの様子. 実際の RGB-D センサは上方に取り付けられており，仮想的に視点を変えているため，ものが重なっている部分は影としてデータが存在しない．見やすさの都合でうさぎ模型を囲うフレームと中心点を強調表示している．テンプレート (b) に比べ，リアルタイムに入力されるデータ (a) では，向きが異なるためデータが大きく欠損しているにもかかわらずトラッキングできている．

[1] 早稲田大学大学院 先進理工学研究科
電気・情報生命専攻, 東京都新宿区大久保 3-4-1, 〒 169-0072
[a)] ksk-uym@ruri.waseda.jp
[b)] masato.inoue@eb.waseda.ac.jp

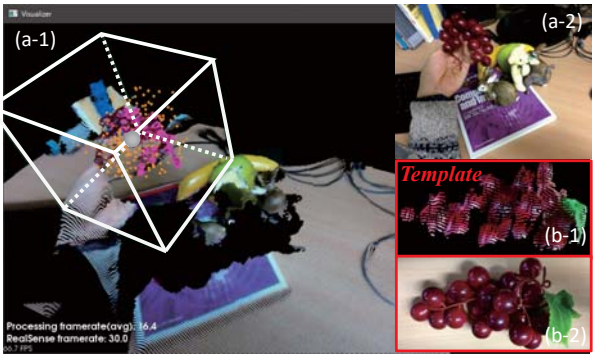

図 2 大きく変形する物体のトラッキングの様子. テンプレート学習時 (b-1，b-2) とぶどうの模型の房の形や見え方が異なる (a-1，a-2) にもかかわらず，トラッキング可能である．

る．一方，トラッキング対象の物体が事前にわからず，物体について最初のフレームデータのみを使ったり，トラッキング中にテンプレートを更新したりする手法にはいくつかのアプローチが提案されている．Point Cloud Library（PCL）[21] では，1 フレームのテンプレート物体の深度画像を利用し，Iterative Closest Point（ICP）アルゴリズムに基づきパーティクルフィルタの尤度を与えるトラッキング手法が実装されている．また，後述する領域分割手法 VCCS を使って，計算量を減らすことでリアルタイムに精度良くトラッキングする手法も提案されている [19]．ただし，基本的な ICP による手法は原理上オクルージョンやデータの乱れなどにはロバストでない．モデルなしで変形する物体をトラッキングする手法としては，対象物体の点群を均一にダウンサンプリングして特徴量を算出して尤度とし，精度とロバストさを両立する手法 [16] やグラフ構造を持った混合正規分布モデル（GMM）を用いて，前フレームとの位置合わせとモデルの更新を行いロバストとする手法 [12] が提案されている．しかし，これらはモデルの更新を常に行うために一度大きく点群が変化し検出に失敗するとその後物体を見失ってしまう可能性がある．

2. 提案手法

本手法では，トラッキングの枠組みとして，スパース表現によるオクルージョンやノイズにロバストな手法である L1 Tracker[17] を活用し，これを 3 次元空間に拡張している．3 次元空間への拡張を行うため，物体表面上の点群の 3 次元特徴量をテンプレートとして用いるが，比較のための点群の選び方や最適化計算の演算量増大の点で工夫が必要である．提案手法では，入力された点群をあらかじめ表面特徴により小領域に分割し，小領域が対象物体のテンプレートの特徴と一致しているかを調べることによりリアルタイムの処理を可能とした（図 **3**）．それだけでなく，局所的な表面の特徴を抽出し利用しているため，学習時に見えていなかった面や物体の大きな変形を許容しながらトラッキングすることが可能である．

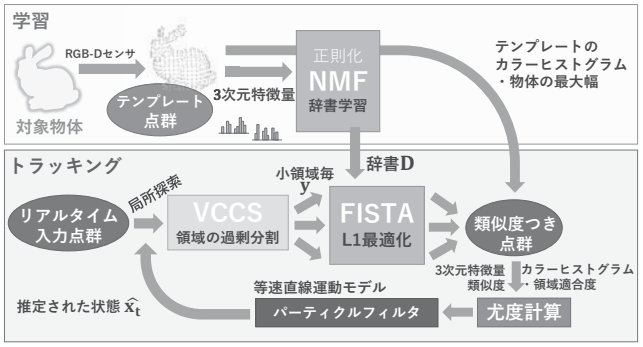

図 3 処理の流れ

2.1 節では，L1 Tracker の手法について概観し，3 次元特徴量への拡張を考える．2.2 節では，対象物体の学習に NMF を用いた対象物体を代表する 3 次元特徴量を抽出する方法について解説する．2.3，2.4 節において，表面特徴や空間的な制約により領域を分割する VCCS を利用し，入力された点群をあらかじめ領域分割することで，小領域とテンプレートの 3 次元特徴量の L1 最適化比較回数を削減する方法や，そこで使用する高速な最適化手法について述べる．2.5 節では，点ごとに与えられた 3 次元特徴量の類似度や色情報を基に，物体のトラッキングに必要となる尤度設計について説明する．

2.1 L1 Tracker

L1 Tracker[17] は，映像においてトラッキングしたいテンプレートをスパースな表現によって表すことにより，L1 最適化を用いて対象物体をトラッキングする手法である．本研究では，3 次元特徴量に対してこの枠組みを適用するが，まず簡単のため，映像における定式化を説明し，2.1.4 で 3 次元特徴量を利用する場合の拡張手法について述べる．

2.1.1 トラッキング対象のスパース表現

トラッキングしたい対象物体の d 画素のテンプレートパッチ画像を 1 列に並べて $\mathbf{t}_k \in \mathbb{R}^d$ とする．このテンプレートの n 個の集合を $\mathbf{T} \equiv [\mathbf{t}_1, \ldots, \mathbf{t}_n] \in \mathbb{R}^{d \times n}$ とする．ただし，n は d よりも十分に小さいものとする. このとき，観測される対象物体の画像 $\mathbf{y} \in \mathbb{R}^d$ は，$\mathbf{T}$ を使って次のように表せる．

$$\mathbf{y} \approx \mathbf{T}\mathbf{a} = a_1\mathbf{t}_1 + a_2\mathbf{t}_2 + \cdots + a_n\mathbf{t}_n \qquad (1)$$

ここでは，$\mathbf{a} \equiv (a_1, a_2, \ldots, a_n)^\top \in \mathbb{R}^n$ を対象物体の係数ベクトルとよぶ．

実際観測される映像にはオクルージョンやノイズが部分的に物体画像上に存在する．これらを考慮するため，文献 [23] での枠組みを利用すると，単位行列 $\mathbf{I} \in \mathbb{R}^{d \times d}$ を用いて観測される画像 $\mathbf{y}$ は次のような関係として書くことができる．

$$\mathbf{y} = [\,\mathbf{T},\ \mathbf{I}\,] \begin{bmatrix} \mathbf{a} \\ \mathbf{e} \end{bmatrix} \equiv \mathbf{B}\mathbf{w}, \quad \text{s.t. } \mathbf{w} \geq 0 \qquad (2)$$

ここで，$\mathbf{e} \equiv (e_1, e_2, \ldots, e_d)^\top \in \mathbb{R}^d$ は，オクルージョンやノイズを表す係数ベクトルである．また，$\mathbf{B} \equiv [\,\mathbf{T},\ \mathbf{I}\,] \in \mathbb{R}^{d\times(n+d)}$，$\mathbf{w}^\top \equiv [\,\mathbf{a},\ \mathbf{e}\,] \in \mathbb{R}^{n+d}$ として定義する．$\mathbf{w}$ は非負値の係数ベクトルである．本論文では，≥ 0 はベクトルや行列の各成分が非負値であることを示す．

2.1.2 L1 最適化による類似度の算出

式 (2) を連立方程式として考えると，$d < n + d$，すなわち式の数よりも変数が多いために解 $\mathbf{w}$ は一意に定まらない．しかしここで，ノイズやオクルージョンは，画像中のわずかな場所でしか発生しないものとすれば，$\mathbf{e}$ はスパースと考えることができ，$d \gg n$ の関係から解 $\mathbf{w}$ もスパースである．解 $\mathbf{w}$ のスパース性を仮定できるため，式 (2) を L1 最適化によって解くことができる．

$$\hat{\mathbf{w}} \equiv \arg\min_{\mathbf{w}} ||\mathbf{y} - \mathbf{B}\mathbf{w}||_2^2 + \lambda||\mathbf{w}||_1 \tag{3}$$

ただし，$\hat{\mathbf{w}}$ は最小化を与える解で，λ は正則化パラメータである．

この性質を使えば，観測された画像の候補 $\mathbf{y}$ がどれだけ対象物体のテンプレートに近いかをスパース度合いによって表すことが可能になる．たとえば，テンプレートと同じ物体が写っているような場合，ほとんどは対象物体のベクトル $\mathbf{a}$ の成分で表現することができ，またノイズやオクルージョンを表す $\mathbf{e}$ もわずかな成分のみを使うことになるために $\mathbf{w}$ はスパースなベクトルになるはずである．一方，テンプレートとは異なるものが写っている場合は，$\mathbf{a}$ の成分ではうまく表現しきれず，$\mathbf{e}$ のさまざまな画素を使って表現しなければならないために，$\mathbf{w}$ はスパースとならない．

2.1.3 パーティクルフィルタ

対象物体の追跡にはパーティクルフィルタ [7] を用いる．Markov 性を満たす直接観測できない時刻 t の内部状態 $\mathbf{x}_t$ を時刻 t までの観測値 $\mathbf{z}_{1:t} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{t-1}, \mathbf{z}_t\}$ から推定する手法であり，ランダムサンプリングにより直前の事後分布 $p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})$ から事前分布 $p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$ やそこから得られる事後分布 $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ を近似することができる．時刻 t の事前分布 $p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$ は $\mathbf{x}_t$ の Markov 性の仮定から

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})\mathrm{d}\mathbf{x}_{t-1} \tag{4}$$

となり，時刻 $t-1$ の事後分布 $p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})$ および，システムモデルから得られる状態遷移確率 $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ によって与えられる．

観測値 $\mathbf{z}_{1:t}$ から $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ を直接推定することは難しいため，これを Bayes の定理により，尤度 $p(\mathbf{z}_t|\mathbf{x}_t)$ と事前分布 $p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$ の積として表現すると，

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) = \frac{p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{z}_{1:t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1})} \tag{5}$$

として逐次的に更新できる．パーティクルフィルタでは，この事後分布 $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ を有限 N 個の粒子 $\{\mathbf{x}_t^i\}_{i=1,\ldots,N}$ と重み w_t^i を使って近似する．提案分布 $\pi(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t})$ から粒子 $\mathbf{x}_t^i$ をサンプルするとすると，重み w_t^i は次式によって計算される．

$$w_t^i \propto w_{t-1}^i \frac{p(\mathbf{z}_t|\mathbf{x}_t^i)p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)}{\pi(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t})} \tag{6}$$

今回用いたブートストラップフィルタでは，さらに提案分布 $\pi(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t}) = p(\mathbf{x}_t|\mathbf{x}_{t-1})$ の状態遷移確率とすることで，重みは結局 $w_t^i = p(\mathbf{z}_t|\mathbf{x}_t^i)$ として尤度を使うことができる．この方法では，粒子を状態遷移確率から容易にサンプリングできる利点もある．

本研究では，等速直線運動モデルを仮定した 3 次元空間の位置での追跡を考える．時刻 t での対象物体の位置 $\mathbf{p}_t \equiv (x_t, y_t, z_t)^\top$，速度 $\mathbf{s}_t \equiv (\dot{x}_t, \dot{y}_t, \dot{z}_t)^\top$ とすると，状態ベクトル $\mathbf{x}_t = [\mathbf{p}_t^\top, \mathbf{s}_t^\top]^\top$ とおける．システムノイズ $\mathbf{v}_t$ を平均 $\mathbf{0}$，共分散 $\mathbf{\Sigma}$ の正規分布に従うものとする．システムモデルは，次のように定義する．

$$\mathbf{x}_t \equiv \mathbf{G}\mathbf{x}_{t-1} + \mathbf{v}_t^i \tag{7}$$

$$\mathbf{G} \equiv \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{v}_t \sim N(\mathbf{0}, \mathbf{\Sigma})$$

時刻 t における推定値 $\hat{\mathbf{x}}_t$ は，w_t^i を正規化した重み W_t^i による重み付き平均によって求めた．

$$\hat{\mathbf{x}}_t = \sum_{i=1}^{N} W_t^i \mathbf{x}_t^i \tag{8}$$

2.1.4 L1 Tracker の点群 3 次元特徴量への拡張

L1 Tracker を 3 次元に拡張することを考えると，たとえば深度画像をそのままテンプレートとして利用することも可能だが，画像では物体の位置・姿勢変化による見え方の変化を考慮する処理が必要で検出も困難となり，3 次元空間での処理の利点が得られず真の意味と拡張にはならない．文献 [3] では，直方体のパーティクルとしそれを分割した領域の特徴量をそれぞれ算出することで 3 次元へ拡張し，深度動画上でのトラッキングを行っている．本提案手法では，点群処理で 3 次元空間の表面構造を比較するために設計された 3 次元特徴量と 3 次元的な領域分割を用いることで L1 Tracker の枠組みを導入した．3 次元特徴量は，物体の姿勢変化，スケール変化，点群密度変化などにロバストに設計されているため，物体と RGB-D センサとの位置関係が変わって得られる点群が変化しても，同一物体を検出することが可能になる．3 次元特徴量の中でもある点の周囲の表面特徴を記述するのに利用される局所特徴量を扱い，Fast Point Feature Histogram（FPFH）および Rotation-Invariant Feature Transform（RIFT）の 2 つの表面特徴量を 1 つのヒストグラムとして利用した．領域分割については，2.3 で述べる．

Fast Point Feature Histogram（FPFH）

FPFH[20] は，PFH[22] を高速化した特徴量で，注目点の法線ベクトルとその周囲の点群の法線ベクトルを比較しヒストグラム化した，表面形状を記述できる特徴量である．ここでは，任意の表面上の点を $\boldsymbol{p}$ として表記している．ある点の組 $\boldsymbol{p}_i$ と $\boldsymbol{p}_j$ に関して，それぞれの法線ベクトル $\boldsymbol{n}_i, \boldsymbol{n}_j$ を推定する．このとき，次の $\alpha_f, \phi_f, \theta_f$ を計算し，それぞれの値を 11 分割してビンに割り当て，ヒストグラムを組み合わせることで 33 次元のヒストグラムとする．このヒストグラムは Simplified Point Feature Histogram(SPFH) と呼ばれ，任意の点 $\boldsymbol{p}_i$ の SPFH を $\boldsymbol{h}_{\mathrm{SPFH}}(\boldsymbol{p}_i)$ と書く，

$$\begin{aligned} \alpha_f &\equiv \boldsymbol{v}\cdot\boldsymbol{n}_j \\ \phi_f &\equiv \frac{\boldsymbol{u}\cdot(\boldsymbol{p}_j-\boldsymbol{p}_i)}{||\boldsymbol{p}_j-\boldsymbol{p}_i||} \\ \theta_f &\equiv \arctan(\boldsymbol{w}\cdot\boldsymbol{n}_j, \boldsymbol{u}\cdot\boldsymbol{n}_j) \end{aligned} \tag{9}$$

ただし，$\boldsymbol{u} \equiv \boldsymbol{n}_i$，$\boldsymbol{v} \equiv (\boldsymbol{p}_j - \boldsymbol{p}_i) \times \boldsymbol{u}$，$\boldsymbol{w} \equiv \boldsymbol{u} \times \boldsymbol{v}$ とする．

ある注目点 $\boldsymbol{p}_q$ に対して，その周囲半径 R_f の球内にある K_f 個の点群 $\{\boldsymbol{p}_i\}_{i=1,\ldots,K_f}$ を考える．$\boldsymbol{p}_q$ と $\boldsymbol{p}_i$ の K_f 個の組に対して，$\boldsymbol{h}_{\mathrm{SPFH}}(\boldsymbol{p}_q)$ を求める．さらに，これら近傍の点 $\{\boldsymbol{p}_i\}_{i=1,\ldots,K_f}$ それぞれの $\boldsymbol{h}_{\mathrm{SPFH}}(\boldsymbol{p}_i)$ を求め，次式によって $\boldsymbol{h}_{\mathrm{FPFH}}(\boldsymbol{p}_q)$ が与えられる．

$$\boldsymbol{h}_{\mathrm{FPFH}}(\boldsymbol{p}_q) \equiv \boldsymbol{h}_{\mathrm{SPFH}}(\boldsymbol{p}_q) + \frac{1}{K_f}\sum_{i=1}^{K_f}\frac{1}{||\boldsymbol{p}_q-\boldsymbol{p}_i||}\cdot\boldsymbol{h}_{\mathrm{SPFH}}(\boldsymbol{p}_i) \tag{10}$$

ただし，$\boldsymbol{p}_q\boldsymbol{p}_i$ 間の距離を $||\boldsymbol{p}_q - \boldsymbol{p}_i||$ とする．

Rotation-Invariant Feature Transform（RIFT）

RIFT[15] は，画像で用いられる SIFT を拡張し 3 次元空間で扱えるようにした物体表面のテクスチャを検出する，回転に不変なヒストグラム特徴量である．ある注目点 $\boldsymbol{p}_q$ に対して，その周囲半径 R_r の球内にある K_r 個の点群 $\{\boldsymbol{p}_i\}_{i=1,\ldots,K_r}$ を考える．$\boldsymbol{p}_i$ それぞれについて，接平面の輝度値の変化が最大になる方向を求める．このとき，$\boldsymbol{p}_q\boldsymbol{p}_i$ 間の距離 d_r と（$\boldsymbol{p}_q$ から $\boldsymbol{p}_i$ へ向かう方向を基準とする）輝度値変化が最大となる方向 θ_r を，それぞれ均等に分割し，当てはまるヒストグラムのビンに K_r 個の点すべてを割り当てる．今回，距離 4 分割，方向 8 分割の 32 次元の特徴量 $\boldsymbol{h}_{\mathrm{RIFT}}(\boldsymbol{p}_q)$ とした．

利用した 3 次元特徴量の構成と L1 Tracker への適用

これら $\boldsymbol{h}_{\mathrm{FPFH}}(\boldsymbol{p}_q) \in \mathbb{Z}^{33}$ と $\boldsymbol{h}_{\mathrm{RIFT}}(\boldsymbol{p}_q) \in \mathbb{Z}^{32}$ を組み合わせた次式で与えられる特徴量 $\mathbf{y}_q \in \mathbb{R}^{65}$ を 3 次元特徴量として提案手法で用いた．

$$\mathbf{y}_q \equiv \left[\boldsymbol{h}_{\mathrm{FPFH}}(\boldsymbol{p}_q)^\top,\ \delta\cdot\boldsymbol{h}_{\mathrm{RIFT}}(\boldsymbol{p}_q)^\top\right]^\top \tag{11}$$

ただし，δ は 2 つのヒストグラムの影響度を調整するパラメータである．

L1 Tracker における画像パッチ $\mathbf{t}_q$ の代わりに定義した $d = 65$ の 3 次元特徴量ヒストグラム $\mathbf{y}_q$ を利用することで，点群の表面構造を比較する．

2.2 NMF によるテンプレートの辞書学習

オリジナルの L1 Tracker では，画像をテンプレートとしていたためにそのまま複数枚の画像をテンプレートとして利用することができたが，3 次元特徴量をテンプレートとするとテンプレート点群中に多数の点が含まれ，そのままテンプレートとして用いると $d \gg n$ のスパースな表現にできない．点群の場合，ある点の周囲には似た特徴量をもった点が多数存在しているため，テンプレートを代表する少数の基底からなる辞書を抽出することできれば，これを利用することができる．

辞書の抽出には非負値行列因子分解（NMF）を用いた．NMF は非負値行列を 2 つの非負値の行列の積に分解する手法であり，スパースな行列を得られることから，特徴を辞書にうまく抽出できることが知られている．テンプレート点群中に含まれる点の 3 次元特徴量 $\mathbf{y}_q \in \mathbb{R}^d$ を m 個並べた行列 $\mathbf{Y} \equiv [\mathbf{y}_1, \ldots, \mathbf{y}_m] \in \mathbb{R}^{d\times m}$ をデータとして，非負行列 $\mathbf{D} \equiv [\mathbf{d}_1, \ldots, \mathbf{d}_n] \in \mathbb{R}^{d\times n}$ および $\mathbf{H} \in \mathbb{R}^{n\times m}$ の積として分解する．辞書 $\mathbf{D}$ に含まれる $\mathbf{d}_k \in \mathbb{R}^d$ は抽出されたテンプレートを代表する 3 次元特徴量と考えることができる． elastic net 様の正則化を加えて，$\mathbf{Y} \approx \mathbf{DH}$ として次の最小化問題を解く．

$$\begin{aligned} \min_{\mathbf{D},\mathbf{H}\geq 0} &\frac{1}{2}||\mathbf{Y}-\mathbf{DH}||_F^2 + \alpha\gamma||\mathbf{D}||_{\ell 1} + \alpha\gamma||\mathbf{H}||_{\ell 1} \\ &+ \frac{\alpha(1-\gamma)}{2}||\mathbf{D}||_F^2 + \frac{\alpha(1-\gamma)}{2}||\mathbf{H}||_F^2 \end{aligned} \tag{12}$$

ただし，$||\cdot||_{\ell 1}$ は行列成分ごとの L1 ノルム，$||\cdot||_F$ はフロベニウスノルムであり，α は正則化パラメータ，γ は L1 および L2 ノルムの割合を決めるパラメータである．

初期値には NNDSVD[4] を利用し，分解のアルゴリズムには greedy coordinate descent 法 [8] を用いることで高速に学習を行う．

テンプレート $\mathbf{T}$ の代わりに，辞書 $\mathbf{D}$ を用いて，$\check{\mathbf{B}} \equiv [\,\mathbf{D},\ \mathbf{I}\,] \in \mathbb{R}^{d\times(n+d)}$ と再定義し，最適化問題 (3) を解くことで物体テンプレートと類似の点群か判断可能となる．

2.3 VCCS による 3 次元領域分割

Voxel Cloud Connectivity Segmentation（VCCS）[18] は，3 次元空間で計算量コストの大きな領域分割や物体認識などを行う前に表面の法線ベクトルの変化や空間的な距離，色などの違いによって過剰に小領域 (supervoxel) へ分割し，その後の処理を軽減する領域分割手法 (**図 4**) である．これによって，物体の判別などはできないが，物体の境界などが予め分割された状態にすることができる．VCCS は深度画像ではなく 3 次元空間での処理でありながらもオン

図 4　VCCS による領域分割

ライン処理可能な効率的なアルゴリズムとなっており，深度画像を扱う場合に比べ，空間的な接続を考慮した正確な領域分割ができる．提案手法では最適化による物体検出の回数を減らすため，VCCS により領域分割をおこなった．

VCCS では，まず 3 次元空間上で点群を R_{voxel} で等間隔に分割しボクセルを作成し，ボクセルの隣接関係からグラフを作成することで 3 次元空間でのつながりを考慮する．ここでは，ボクセルの面，辺，頂点のいずれかが接しているボクセルを隣接しているとする．間隔 R_{seed} でシードとなるボクセルを定める．次に定義する距離 D による局所的な k-means 法を適用し，隣接グラフ内での探索によって region growing を行うために 3 次元空間でのつながりが維持され，適切な領域分割が行われる．

分割に用いる距離 D は，RGB 空間で正規化されたユークリッド距離 D_c，シードの間隔 R_{seed} で正規化された空間的距離 D_s，および表面の法線ベクトルの差 $D_n \equiv 1 - \boldsymbol{p}_j \cdot \boldsymbol{p}_i$ を用いて次式と定義する．

$$D \equiv \sqrt{\eta D_c^2 + \mu \frac{D_s^2}{3R_{\mathrm{seed}}^2} + \xi D_n^2} \tag{13}$$

このとき，η，μ，ξ はそれぞれ色空間距離，空間的距離，法線ベクトルの差の影響度を調整するパラメータである．

適用範囲の限定

VCCS は比較的高速な手法であるが，十分な細かさで毎フレーム全点群に処理を施すとリアルタイムの動作が難しいため，予測された物体位置の周囲の点群のみに対して適用した．これは推定された 1 フレーム前の状態 $\hat{\mathbf{x}}_{t-1}$ の位置 $\hat{\mathbf{p}}_{t-1}$ に速度 $\hat{\mathbf{s}}_{t-1}$ を加え，1 フレーム分進めた予測位置 $\tilde{\mathbf{p}}_t$ を中心として．半径 $R_v \equiv \zeta \cdot l_{\max}$ の球内の点群を探索し適用範囲とすることで実現している（**図 5**）．ただし，$l_{\max}$ は物体テンプレートを直方体で覆ったときの最大幅であり，ζ はその係数パラメータである．

2.4　FISTA による L1 最適化と類似度算出

多数の 3 次元特徴量の類似度を求めるには，多くの最適化を行う必要があり，リアルタイム処理が必要な本研究では，最適化問題 (3) の高速化が欠かせない．今回，最適化すべき関数は，$F(\mathbf{w}) = f(\mathbf{w}) + g(\mathbf{w})$ として，微分可能な凸関数 $f(\mathbf{w}) = ||\mathbf{y} - \check{\mathbf{B}}\mathbf{w}||_2^2$ と微分不可能な点を含むが近接写像 prox 作用素を定義可能な関数 $g(\mathbf{w}) = \lambda||\mathbf{w}||_1$ の和として表現可能な問題であるため，近接勾配法を用いることができる．ここでは，近接勾配法の収束を高速化した FISTA[2] を用いた．

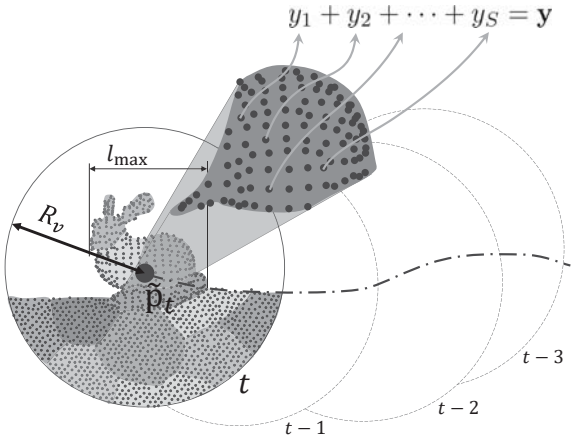

図 5　探索範囲の限定と小領域のサンプリング

サンプリングによる類似度算出

処理の高速化のため，1 つの点の 3 次元特徴量の類似度を算出するのではなく，VCCS によって分割されたある小領域が物体テンプレートの一部であるかを類似度として算出したい．そこで，テンプレートと類似の点であれば複数足し合わせても $\hat{\mathbf{w}}$ は依然スパースとなるはずであるので，小領域中の点群をランダムに S 個サンプルして足し合わせることによって，小領域にテンプレートと類似した特徴が含まれているかを一度の最適化で算出可能となる（図 5）．その小領域中の点の特徴量 $\mathbf{y}_q$ を足し合わせ，小領域全体の特徴量 $\mathbf{y}$ を計算する．

$$\mathbf{y} \equiv \sum_{q=1}^{S} \mathbf{y}_q \tag{14}$$

毎フレーム，あらかじめすべての小領域について $\mathbf{y}$ を計算しておき，尤度の計算に利用する．

2.5　尤度関数の設計

パーティクルフィルタの利点の一つは，明示的な観測モデルを与える必要がないことであり，テンプレートにどれだけ観測された値が近いかという尤度 $\mathcal{L} \equiv p(\mathbf{z}_t|\mathbf{x}_t^i)$ を定義できれば良い．尤度関数は以下に述べる 3 次元特徴量，物体のサイズ，色ヒストグラムの 3 つの類似度を基に算出するよう設計した．処理の大枠としては，パーティクル $\mathbf{x}_t^i$ の位置 $\mathbf{p}_t^i$ を中心とする半径 $R_l = \nu \cdot l_{\max}/2$ の球内にある点群それぞれのあらかじめ求めておいた類似度を加算して，パーティクル $\mathbf{x}_t^i$ に対する尤度を求めた．この方法ではパーティクルごとに最適化計算をする必要がなくなり高速化できる．ν は球の大きさを調整するパラメータであるが，基本的に半径 R_l の球は物体テンプレートがちょうど収まる大きさであり，球内の点群にテンプレートが含まれ

るほど類似度が高くなるように設計することで，できるだけ正確にトラッキングできるようにする（図 **6**）．

3 次元特徴量の類似度

テンプレートとの類似度を表現するため，$\hat{\mathbf{w}}$ から得られる $\hat{\mathbf{a}}$ による次式を用いる．小領域ごとにあらかじめ残差 r を計算しておく．

$$r \equiv ||\mathbf{y} - \mathbf{D}\hat{\mathbf{a}}||_2 \tag{15}$$

物体サイズの考慮

画像では，物体の位置が変わると物体の見かけの大きさが変わってしまうが，3 次元空間では剛体であれば常に一定である．そのため，物体はテンプレート学習時に得られる直方体の最大幅 $l_{\max}$ を基準に半径 R_l の球内に収まるはずであり，物体の一部である小領域もこの中に収まっているはずである．この仮定をもとに小領域がどの程度球内に収まっているかという値を求め，s で表す．小領域の点の個数を N_s，そのうち球内に含まれる個数を o とする．

$$s \equiv \frac{o}{N_s} \tag{16}$$

色ヒストグラムによる類似度

RGB 色空間から HSV 色空間に変換し，色相と彩度をそれぞれ 21 分割と 7 分割し $N_h = 147$ ビンとする．点群の点を当てはまるビンに割りあて，その数を正規化したものを色ヒストグラムとして利用した．テンプレートの色ヒストグラムを $\bar{\mathbf{c}} \equiv (\bar{c}_1, \ldots, \bar{c}_{N_h})^\top$，ある小領域の色ヒストグラムを $\mathbf{c} \equiv (c_1, \ldots, c_{N_h})^\top$ とすると，類似度は Bhattacharyya 係数 h によって表すことができる．

$$h \equiv \sum_{h=1}^{N_h} \sqrt{\bar{c}_h c_h} \tag{17}$$

尤度関数

まず，3 つの類似度を小領域全てで計算しておく．同じ小領域内の点はすべて同じ類似度を持っていることになる．$\mathbf{p}_t^i$ を中心とする半径 R_l の球内に含まれる N_p 個の点すべての類似度について，それらの合計を 1 つの変数 b として以下のように計算する．

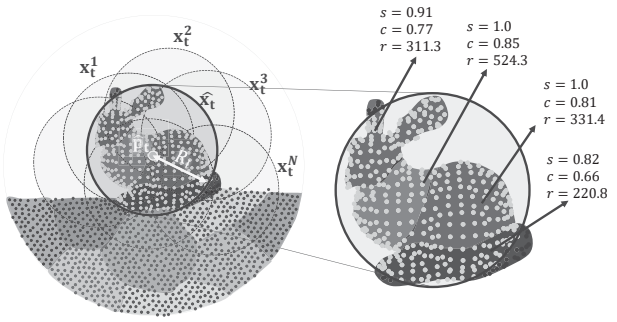

図 6 パーティクルを中心とする半径 R_l の球による探索と各小領域における類似度の例

$$b \equiv \sum_{p=1}^{N_p} \left\{ \frac{1}{r_p^\phi} \cdot s_p^\psi \cdot h_p \right\} \tag{18}$$

ただし，ϕ，ψ は，それぞれ 3 次元特徴量と物体サイズに関する項の冪数であり，各類似度の影響を調整するパラメータである．b は大きな数値であるほど類似度が高いことを意味するため，これを尤度関数 $\mathcal{L}$ として定義する．

$$\mathcal{L} \equiv \frac{1}{1 + \kappa \cdot \exp(-\tau \cdot b)} \tag{19}$$

κ，τ はロジスティック関数のパラメータである．

物体上の点群の推定

推定された $\hat{\mathbf{x}}_t$ の $\hat{\mathbf{p}}_t$ を中心として，半径 R_l の球を考え，その中に含まれる小領域について，$s > \sigma$ かつ $c > \chi$ の領域を物体上の点群とする．σ，χ は閾値である．

3. 実験

3.1 実装およびハードウェア

プログラムは，Point Cloud Library（PCL）[21] を用いて点群操作や表示処理を行っている．パーティクルフィルタ部分や小領域ごとの類似度の算出等は並列化している．

RGB-D センサには，コード化されたパターンを照射することで 0.2m〜1.5m の範囲の深度情報を得る Intel RealSense SR300 を用いた．今回は深度画像解像度 640×240 とカラー画像解像度 320×180 で入力している．また，実験は Core i7-3770 CPU を搭載した PC 上で行った．

3.2 実験で用いたパラメータ

実験時のパラメータは次の通りである．パーティクルフィルタのパーティクル数 $N = 500$，辞書の基底数 $n = 8$，NMF の正則化パラメータ $\alpha = 0.3$，L1・L2 ノルムの正則化項の比率を変化させるパラメータ $\gamma = 0.2$，FISTA の正則化パラメータ $\lambda = 1.7 \times 10^{-5}$，小領域のサンプリング点の数 $S = 20$，3 次元特徴量の影響比率 $\delta = 50.0$，VCCS のシード間隔 $R_{\text{voxel}} = 15$[cm]，解像度 $R_{\text{seed}} = 1.2$[cm]，距離を定義するパラメータ $\eta = 0.1$，$\mu = 0.5$，$\xi = 1.2$，探索範囲を決めるパラメータ $\zeta = 1.1$，尤度関数の球直径を決めるパラメータ $\nu = 0.9$，尤度関数の 3 次元特徴量と物体サイズの影響度を変える冪数 $\phi = 0.5$，$\psi = 1.3$，ロジスティック関数のパラメータ $\kappa = 1000$，$\tau = 10.5$，物体上の点群を判定する閾値 $\sigma = 0.7$，$\chi = 0.55$ とした．

3.3 トラッキングの性能評価

まず，はじめの 1 フレームをテンプレートとして学習する平均時間を算出した（表 1）．学習を高速化するため，取得された点群を 1.5[mm] 間隔にダウンサンプリングしてあり，また，数値には 3 次元特徴量の計算時間も含まれている．

続いて，学習したテンプレートで，剛体と非剛体の物体

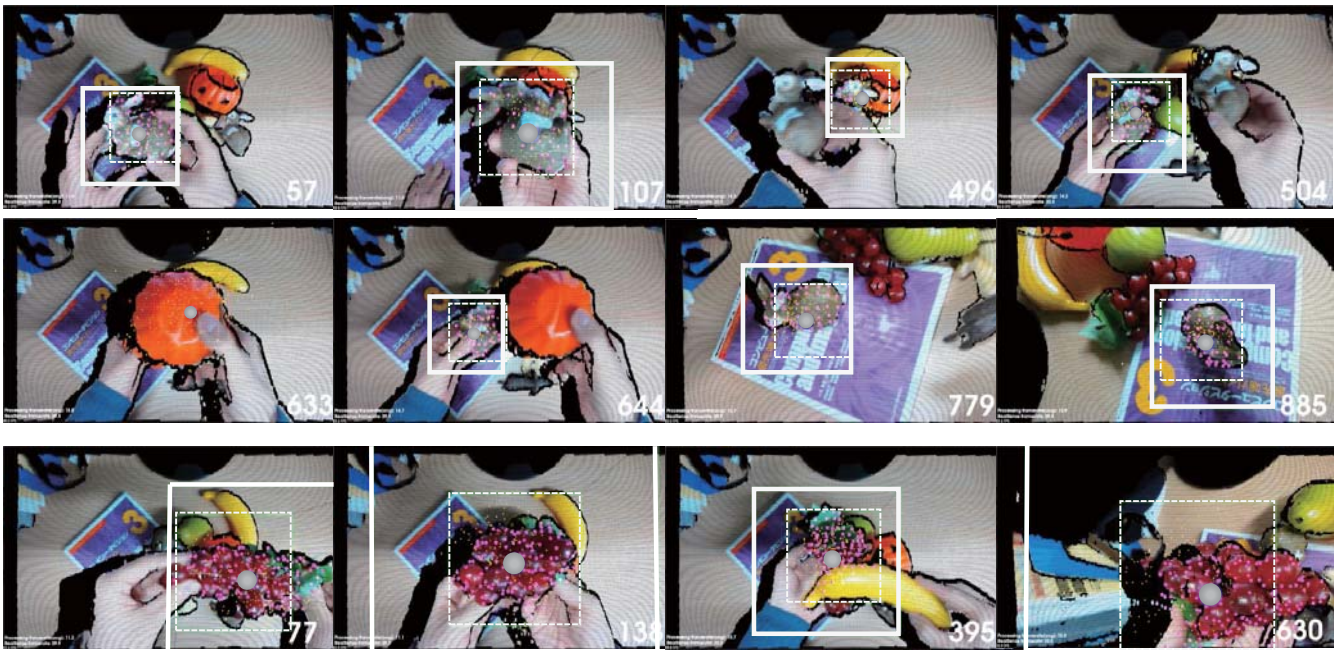

図 7 うさぎ模型（上 2 段）およびぶどう模型（下段）のトラッキング（右下の数字はフレーム番号，丸は推定された位置であり，模型を囲う立方体フレームは強調表示している）

表 1 テンプレート点群の平均学習時間 (Mean±SD)

	テンプレート中の点の数	学習時間 [ms]
うさぎ模型	1284.1 ± 11.6	823.6 ± 12.4
ぶどう模型	1709.2 ± 50.1	1102.1 ± 45.3

に対するトラッキングを行った．表 2 にトラッキング時の平均処理フレームレートを示す．この実験では実際に RGB-D センサからデータを入力しながら計算処理している．なお，わかりやすさのため，図では入力されたすべての点群を表示しているが，表示負荷が存在するため表示しない場合についても計測した．

表 2 トラッキング平均フレームレートの比較 (Mean±SD)

	背景表示あり [fps]	背景表示なし [fps]
うさぎ模型	16.2 ± 1.4	22.8 ± 6.7
ぶどう模型	11.8 ± 1.0	16.3 ± 1.8

図 7 に剛体と非剛体をトラッキングした結果を一部切り出した様子を示す．ここで，物体周囲の立方体は $\hat{\mathbf{x}}_t$ の位置 $\hat{\mathbf{p}}_t$ を中心とし，$l_{\max}$ の大きさを示している．うさぎ模型の剛体について，学習時の姿勢のフレーム 57 だけでなく様々な姿勢に変化させ，学習した面が写っていないフレーム 107 でもトラッキングが維持された．フレーム 496〜504，フレーム 633〜644 はオクルージョンを発生させたが，同じ模型や完全に模型が一度隠れる場合でもトラッキングし続けることができている．フレーム 779 や 885 では，RGB-D センサそのものを動かして，角度を変え近づけている．このときもトラッキングを安定して続けることができた．

また，下段に示した非剛体の大きく変形するぶどうの模型では，学習時に近いフレーム 77 の状態からフレーム 138 のように物体を掴んで大きく変形させた状態でもトラッキングできた．フレーム 395 では，部分的なオクルージョンを発生させているが，きちんとぶどうの部分を検出できている．うさぎ模型同様に，フレーム 630 では視点を変えてもトラッキングできている．

トラッキングが失敗するケースについて，まず平面で構成され表面特徴が乏しい箱のような形状が課題として挙げられる．トラッキングが途中で失敗してしまうケースとしては，等速直線運動を大きく外れる急激な加速が伴う運動があった場合や一定以上の長さで物体が完全に隠れてしまう場合，静止状態など動きがあまり変わらない中で周囲に形状の似た物体がある場合などが挙げられる．

3.4 トラッキングの精度計測

トラッキングの精度を計測するため，ターンテーブル上にうさぎ模型を置き，中心位置座標 $\hat{\mathbf{p}}_t$ を記録したグラフを**図 8** に示す．値はカメラ座標からターンテーブルの鉛直上方を z 軸とする座標系に補正されている．理論的には，x 軸，y 軸は三角関数の軌道を取り，高さは変わらないの

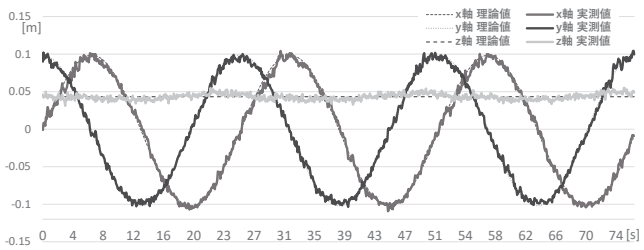

図 8 ターンテーブル上のうさぎ模型の中心位置

で z 軸は一定値であるはずである．実測値と理論値の誤差の絶対値の平均は，x 軸で 3.9[mm]，y 軸で 3.2[mm]，z 軸で 3.3[mm] となった．なお，z 軸の変動にはターンテーブルのたわみも含まれていると考えられる．

4. 結論

本研究の RGB-D センサで得られた点群データに対して 1 フレーム分の深度画像をもとに物体をトラッキングする手法について，大きな姿勢変化や変形があるケースなどでも有効であることを示した．リアルタイムで動作可能であり，実際にロボット等に組み込む応用についても有用な手法となっている．

今後は剛体については検出された物体上の点群からさらに 6 軸の姿勢検出を行うことやより幅広い形状の物体に対して手法を適用できるようにしていく．

謝辞 本研究の一部は科学研究費補助金 (研究課題番号 25120009) の助成を受けたものです．

参考文献

[1] Akkaladevi, S., Ankerl, M., Heindl, C. and Pichler, A.: Tracking multiple rigid symmetric and non-symmetric objects in real-time using depth data, *Proceedings of IEEE International Conference on Robotics and Automation* (2016).

[2] Beck, A. and Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM journal on imaging sciences*, Vol. 2, No. 1, pp. 183–202 (2009).

[3] Bibi, A., Zhang, T. and Ghanem, B.: 3d part-based sparse tracker with automatic synchronization and registration, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1439–1448 (2016).

[4] Boutsidis, C. and Gallopoulos, E.: SVD based initialization: A head start for nonnegative matrix factorization, *Pattern Recognition*, Vol. 41, No. 4, pp. 1350–1362 (2008).

[5] Brachmann, E., Krull, A., Michel, F., Gumhold, S., Shotton, J. and Rother, C.: Learning 6d object pose estimation using 3d object coordinates, *European Conference on Computer Vision*, Springer, pp. 536–551 (2014).

[6] Choi, C. and Christensen, H. I.: RGB-D object tracking: A particle filter approach on GPU, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, pp. 1084–1091 (2013).

[7] Doucet, A., De Freitas, N. and Gordon, N.: An introduction to sequential Monte Carlo methods, *Sequential Monte Carlo methods in practice*, Springer, pp. 3–14 (2001).

[8] Hsieh, C.-J. and Dhillon, I. S.: Fast coordinate descent methods with variable selection for non-negative matrix factorization, *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 1064–1072 (2011).

[9] Jamie Shotton, Andrew Fitzgibbon, A. B. A. K. M. F. R. M. T. S.: Real-Time Human Pose Recognition in Parts from a Single Depth Image, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1297–1304 (2011).

[10] Joseph, T. D. et al.: A Versatile Learning-based 3D Temporal Tracker: Scalable, Robust, Online, *Proceedings of the IEEE International Conference on Computer Vision*, Vol. 1, No. 4 (2015).

[11] Joseph, T. D. and Ilic, S.: Multi-forest tracker: A chameleon in tracking, *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, pp. 1202–1209 (2014).

[12] Koo, S., Lee, D. and Kwon, D.-S.: Incremental object learning and robust tracking of multiple objects from RGB-D point set data, *Journal of Visual Communication and Image Representation*, Vol. 25, No. 1, pp. 108–121 (2014).

[13] Krull, A., Brachmann, E., Michel, F., Ying Yang, M., Gumhold, S. and Rother, C.: Learning Analysis-by-Synthesis for 6D Pose Estimation in RGB-D Images, *Proceedings of the IEEE International Conference on Computer Vision*, pp. 954–962 (2015).

[14] Krull, A., Michel, F., Brachmann, E., Gumhold, S., Ihrke, S. and Rother, C.: 6-dof model based tracking via object coordinate regression, *Asian Conference on Computer Vision*, Springer, pp. 384–399 (2014).

[15] Lazebnik, S., Schmid, C. and Ponce, J.: A sparse texture representation using local affine regions, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 8, pp. 1265–1278 (2005).

[16] Li, S., Koo, S. and Lee, D.: Real-time and model-free object tracking using particle filter with joint color-spatial descriptor, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, pp. 6079–6085 (2015).

[17] Mei, X. and Ling, H.: Robust visual tracking using L1 minimization, *Proceedings of the IEEE International Conference on Computer Vision*, IEEE, pp. 1436–1443 (2009).

[18] Papon, J., Abramov, A., Schoeler, M. and Worgotter, F.: Voxel cloud connectivity segmentation-supervoxels for point clouds, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2027–2034 (2013).

[19] Papon, J., Schoeler, M. and Worgotter, F.: Spatially stratified correspondence sampling for real-time point cloud tracking, *IEEE Winter Conference on Applications of Computer Vision*, IEEE, pp. 124–131 (2015).

[20] Rusu, R. B., Blodow, N. and Beetz, M.: Fast point feature histograms (FPFH) for 3D registration, *Proceedings of IEEE International Conference on Robotics and Automation*, IEEE, pp. 3212–3217 (2009).

[21] Rusu, R. B. and Cousins, S.: 3D is here: Point Cloud Library (PCL), *Proceedings of IEEE International Conference on Robotics and Automation*, Shanghai, China (2011).

[22] Rusu, R. B., Marton, Z. C., Blodow, N. and Beetz, M.: Persistent point feature histograms for 3D point clouds, *Proceedings of 10th Int Conf Intel Autonomous Syst (IAS-10), Baden-Baden, Germany*, pp. 119–128 (2008).

[23] Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S. and Ma, Y.: Robust face recognition via sparse representation, *IEEE transactions on pattern analysis and machine intelligence*, Vol. 31, No. 2, pp. 210–227 (2009).