

多視点映像閲覧のためのデータハンドリングとカメラ位置ズレ補正

志田 全弘[†] 亀田 能成^{†*} 北原 格^{†**}

大田 友一[#] 松田 壮一郎^{#*} 山本 淳一^{‡*} 鈴木 健嗣^{*†}

[†]筑波大学大学院システム情報工学研究科 〒305-8573 茨城県つくば市天王台 1-1-1

+筑波大学計算科学研究センター 〒305-8577 茨城県つくば市天王台 1-1-1

#筑波大学 〒305-8577 茨城県つくば市天王台 1-1-1

‡慶應義塾大学文学部人文社会学科 〒108-8345 東京都港区三田 2-15-45

*独立行政法人科学技術振興機構, CREST 〒332-0012 埼玉県川口市本町 4-1-8

E-mail: †s1520786@u.tsukuba.ac.jp, {kameda,kitahara}@iit.tsukuba.ac.jp

あらまし 児童発達支援や幼児保育のような、多数の被写体が不規則に動き回る環境に適用可能な多視点映像撮影方式と、Bullet-Time 映像を用いた閲覧方式に関する研究開発に取り組んでいる。本稿では、撮影した多視点映像データを効率的にハンドリングする枠組みを紹介する。また、撮影時に被写体の接触により、カメラの位置・姿勢に変動が生じた場合でも、再キャリブレーションをすることなく、Bullet-Time の品質を維持する幾何学的補正法について述べる。

キーワード 多視点映像, データハンドリング, カメラ位置ズレ, Bullet-Time 映像

1. はじめに

本稿では、複数台のカメラで撮影した多視点映像を効率的に計算機に取り込み管理するためのデータハンドリング方式と、Bullet-Time 映像[1]を用いて多視点映像を閲覧する際にカメラの位置ズレの影響を軽減する手法について提案する。

我々は、育児・発達教育のセラピストの熟練度向上や児童とセラピストのインタラクション分析を目的として、複数の高解像度カメラを用いた多視点映像撮影と、それらの映像の直感的な閲覧を可能とする提示システムの研究開発に取り組んでいる[2]。図1に示すように、我々が開発した撮影システムは、小型のアクションカメラ（本システムでは GoPro Hero4 を使用）を育児用サークルのパネルに取り付け、パネル同士を接続することで構成される。以降本稿では、この撮影システムをメディアサークルと呼ぶ。撮影の開始・終了の制御は無線信号によって行うため、ケーブルを配線する必要がない。メディアサークルによって撮影された多視点映像データは、各カメラの記録メディア（本システムでは microSD）に保存される。

多視点映像の閲覧には、映像切替えによって視点移動感を観察者に提示する Bullet-Time 方式を用いる。具体的には、被写体を中心にほぼ等間隔で周囲を取り囲むように複数台のカメラを配置し、同時刻に撮影した多視点画像をカメラの位置関係に応じて順番に切り替える[1]。明智ら[3]は、多視点映像から被写体の3次元情報とカメラの位置・姿勢を同時に推定する Structure from Motion [4](以下 SfM)と注視点の3次元位置推定処理を組み合わせ、動的に注視点の移動が可能な



図1. 撮影システム（メディアサークル）とその一部

Bullet-Time 映像生成法を提案している。また、新たに設定した注視点を中心としたデジタルズーム処理によって、疑似的に視点の前後移動を実現している。本システムでは、明智らの手法を用いて提示映像を生成する。

本研究の映像閲覧を実現するために解決すべき二つの課題について述べる。一つは、多視点映像データから同時刻に撮影したフレームをカメラ台数分抜き出す作業である。Bullet-Time では視点切り替えを行うために、「映像データとカメラ番号の対応関係」が必要で

ある。また各映像データから同時刻に撮影されたフレームを抽出するために「各フレームの撮影時刻もしくは撮影開始時刻」の情報も必要である。一方で、長時間撮影した多視点映像データはデータ量が膨大になるため、計算機に取り込むだけでも長い時間を要する。テイク毎に分割して撮影することにより、各ビデオシーケンスのデータ量は軽減されるが、その場合は、上述した情報に加え、「テイク番号」の情報も必要となる。これらの情報のタグ付けを手動で行うためには多大な手間が必要となる。我々は、カメラの記録メディアを計算機に挿入すれば、自動的に撮影データを区分けしながら計算機に保存する枠組みを構築し、上述した労力を大幅に削減する。

二つ目の課題として、明智らによって開発された Bullet-Time 映像生成法では、SfM で推定したカメラ位置・姿勢情報(カメラキャリブレーションパラメータ)を用いることで注視点の動的な移動を実現しているが、本研究が対象とする多数の被写体が不規則に動き回る撮影環境では、被写体が撮影システムに衝突するケースが頻発し、その都度カメラキャリブレーションパラメータを推定し直す必要がある。しかし、図 2 に示す育児スペースのような空間で撮影された映像には、十分なテクスチャが写っていないケースが多く、結果として、カメラキャリブレーション処理に必要な画像間の対応点情報の取得が困難になる。この対策として、テクスチャ情報を十分有する“位置合わせ用オブジェクト”を撮影空間内に配置することが考えられるが、撮影中そのようなオブジェクトを配置し続けることは現実的ではない。また、育児現場などでは、被写体によっては、目に付きやすいテクスチャが介入作業の妨げになることが懸念される。カメラ位置・姿勢が動く度に位置合わせ用のオブジェクトを配置する対応も考えられるが、その度に介入作業を中断することとなるためやはり現実的ではない。本稿では、簡単なインタラクションにより、カメラの位置・姿勢ズレを補正し、Bullet-Time の映像品質を維持する手法を提案する。



図 2. 育児現場で撮影した多視点映像の一例
(カメラ番号を左上に示す)

2. 関連研究

Ozyesil ら[5]は、SfM における画像間の点の対応付けの推定とそれに基づく相対的なカメラペアの姿勢推定において外れ値が含まれた場合における、安定したカメラの位置・姿勢修正手法を提案している。SfM は同一のシーンから成る 2 次元画像に対応しているカメラの動きを推定することで 3 次元構造を復元する問題である。一般的に SfM は以下の 3 段階から構成される。

- (1)画像間の点の対応付けの推定とそれに基づく相対的なカメラペアの姿勢推定
- (2)カメラモーションの推定、すなわち相対的な姿勢から大域的なカメラの姿勢と位置を推定
- (3)バンドル調整等を用いて再投影誤差を最小化することで推定されたカメラモーションに基づく 3 次元構造の復元

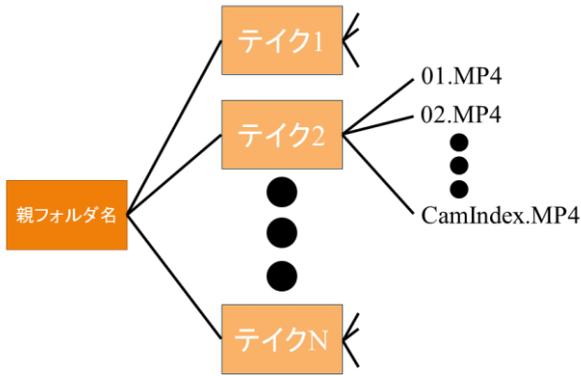
(1)において対応付けを画像全体で行うことは困難であり、加えてカメラの構造上、画像には歪みが生じていることから(1)の相対的なカメラペアの姿勢推定には潜在的に誤差が発生する。そこで Ozyesil らは(2)のカメラ位置推定問題に対し、平行剛性理論を適用することで凸最適化による安定したカメラ位置推定手法を提案している。評価実験では、従来手法と比較し、(1)の外れ値が多い場合に対してより安定したカメラ位置推定を行えることを示している。

しかしこの手法では、カメラモーションを推定し 3 次元構造を復元することを目的としており、評価実験の結果においても、数 10cm 程度のカメラ位置の誤差が生じている。本研究の閲覧システムでは数 cm の誤差で視点切り替え時にブレが生じるため、視点切り替え時の滑らかさの改善を目的とする本研究には適さない。

3. 多視点映像のデータハンドリング

本節では、1 節で紹介した二つ目の課題への取り組みとして、多視点映像の撮影データを効率的に計算機に取り込む手法について述べる。本手法では、図 3 に示す構造を有するフォルダ群を生成し、それに対応するように多視点映像データを保存する。その際、撮影データを検索するために Windows.h の関数 FindFirstFile() を、新規フォルダを作成するために direct.h の関数 mkdir() を使用する。

以下、処理の流れについて述べる。まず最初に、図 4.1~4.3 に示すように、ユーザーが全てのデータが格納される親フォルダ名と計算機に接続している microSD のカメラ番号を入力する。これによって計算機は「映像データとカメラ番号の対応関係」を獲得する。次に、ワイルドカードを指定することで映像拡張子の全てのデータを microSD 内から探索する。この探



分割したテイク数: N, 使用したカメラ台数: CamIndex

図 3. 生成されるフォルダ階層



図 4.1. 親フォルダ名の入力前



図 4.2. 親フォルダ名の入力後

索引結果から撮影映像シーケンスのテイク数分かるため、図 4.4 に示すように、テイク数分のフォルダを作成する（「テイク番号」の獲得）。最後に、図 4.5 に示すように、カメラ番号をファイル名とし、各テイクフォルダに撮影データをコピーする。

本研究では、映像と共に録音された音声情報の波形解析を用いて、テイクフォルダに保存されているテイク毎の多視点映像を同期させる[6]。この結果、Bullet-Time 映像生成に必要な「撮影開始フレーム時刻」を得る。

以上の処理により、大量の多視点映像を計算機に取り込む際のユーザーの負担を減らしつつ、Bullet-Time の生成に必要な情報（「映像データとカメラ番号の対応関係」、「テイク番号」、「各フレームの撮影時刻もしくは撮影開始時刻」）を獲得し、効率的なデータハンドリングを実現する。

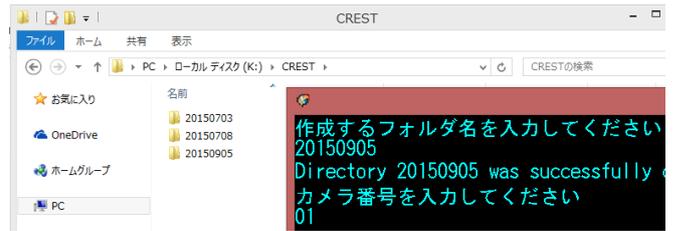


図 4.3. カメラ番号の入力

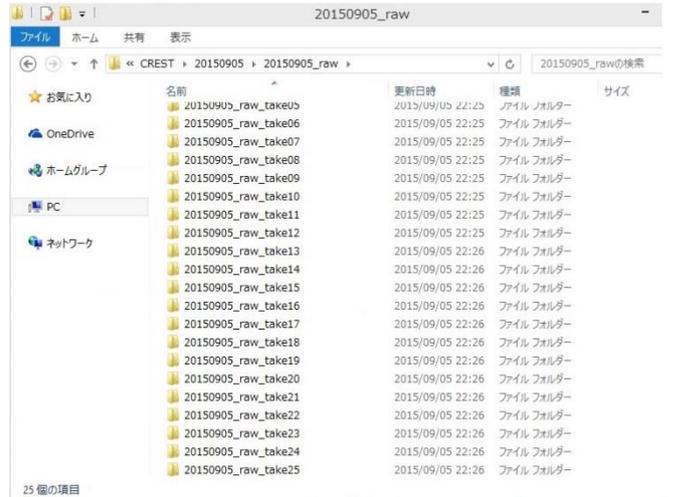


図 4.4. テイク数分のフォルダの作成



図 4.5. カメラ番号をファイル名とし
テイクフォルダ内に保存

4. 映像のズレ補正

本節では、撮影中に被写体が撮影システムに衝突し、カメラの位置・姿勢が変動した場合に、視点切り替えの連続性が損なわれる問題を、簡単なインタラクションによって解決する処理について述べる。

4.1. 注視点の並進移動行列 D の算出

図 5.1 および 5.2 を用いて、カメラの位置・姿勢ズレによる注視点の観測位置の変化について説明する。多視点画像群から動いていないカメラ 2 台を選び、そ

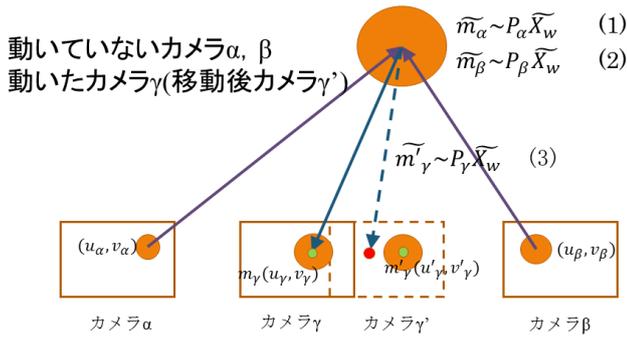


図 5.1. 注視点のズレ量の算出

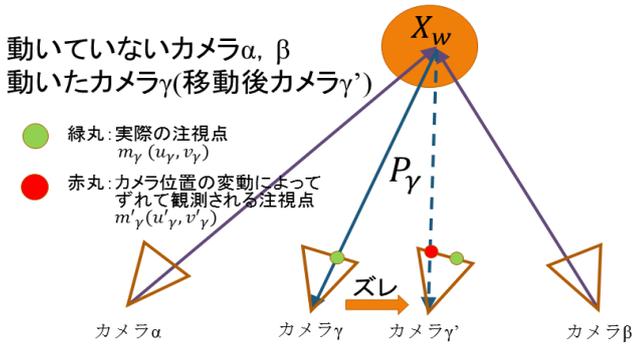


図 5.2. カメラ γ が動く様子

これらの画像上での観測位置情報を用いて注視点の3次元位置を算出する．動いていないカメラ2台をそれぞれカメラ α , カメラ β とすると, 式(1)および(2)によって与えられる連立方程式から, カメラ α , カメラ β で共通して観測される対応点の3次元点 X_w を求める．

$$\begin{aligned} \tilde{m}_\alpha &\sim P_\alpha \tilde{X}_w & (1) \\ \tilde{m}_\beta &\sim P_\beta \tilde{X}_w & (2) \end{aligned}$$

ここで, m_i はカメラ i の注視点の画像座標 (u_i, v_i) を表し, P_i は SfM によって推定されたカメラ i の焦点距離および光学中心を表す内部パラメータ行列とカメラ i の位置・姿勢を表す外部パラメータ行列によって構成される射影変換行列である．また, \tilde{m}_i, \tilde{X}_i は画像座標 m_i および3次元座標 X_i の同次座標を表す．

衝突などにより位置・姿勢がズレたカメラをカメラ γ' とする．このカメラのズレる前の射影変換行列 P_γ を用いて注視点 \tilde{X}_w を画像上に射影した位置 (u', v') は, 式(3)より求めることができる．

$$\tilde{m}'_\gamma \sim P_\gamma \tilde{X}_w \quad (3)$$

注視点が実際に観測されている画像座標を (u, v) とすると, (u', v') と (u, v) の差分により $(\Delta u, \Delta v)$ が求まる．図 5.1 および 5.2 において (u', v') を赤色の丸で表し, (u, v) を緑色の丸で表す．式(4)および(5)に示すように, 撮影画像中の点 (u', v') を $(\Delta u, \Delta v)$ だけ並進移動させる(行列 D をかける)ことにより, カメラ γ' で観測されている注視点 (u', v') をズレがない場合の注視点 (u, v) に移動させることができる．以上の処理により, カメラの位置・姿勢に変動があった場合においても, それによって生じる注視点位置の移動を軽減した閲覧を実現する．

$$D = \begin{pmatrix} 1 & 0 & \Delta u \\ 0 & 1 & \Delta v \\ 0 & 0 & 1 \end{pmatrix} \quad (4)$$

$$\tilde{m}'_\gamma \sim D \tilde{m}_\gamma \quad (5)$$

4.2. テンプレートマッチングによる対応点の推定

前節で述べた行列 D を求めるためには, カメラ γ において注視点が実際に観測されている画像座標 (u, v) (図 5.1 および 5.2 における緑丸) を推定する必要がある．Bullet-Time 映像提示を想定して撮影した多視点映像では, 隣接するカメラ間隔が比較的狭いことを利用し, 画像間での対応点探索処理を用いて観測位置を推定する．

まず, 隣接するズレていないカメラ α で撮影した画像中から, 注視点周辺の領域を矩形で取り出しテンプレートを獲得する．このテンプレートと類似した見え方の領域をズレたカメラ γ で撮影した画像中から探索することで (u, v) を推定する．テンプレートサイズを K [画素] \times K [画素], カメラ位置・姿勢の変動による投影誤差(注視点が実際に観測される位置と射影変換行列で算出される投影点の位置の距離)が垂直/水平成分それぞれに最大 N [画素] とすると, ズレる前の射影変換行列 P_γ を用いて注視点 \tilde{X}_w を画像上に射影した位置 (u', v') から $(2N + K)$ [画素] \times $(2N + K)$ [画素] が探索処理の範囲となる．しかし, 単純なテンプレートマッチングでは注視点以外の領域が検出される場合があるため, 本手法では, 以下に述べるアルゴリズムに基づき, テンプレートマッチングの精度を高める．

K [画素] \times K [画素] の領域を探索用テンプレート画像 $T(u_T, v_T)$ とし, $(2N + K)$ [画素] \times $(2N + K)$ [画素] のテンプレートの探索対象となる画像を $I(u_I, v_I)$ とする．画像 T はカメラ γ' で撮影した画像から (u', v') を中心として切り出し, 画像 I はカメラ α から図 5.1 で表す (u_α, v_α) を中心に切り出す, 式(6)を用いて画像 T と画像 I の正規化相互相関 $R(x, y)$ を求め, R が最大となるときの座標 (u_{match}, v_{match}) を求める．

$$R(x, y) = \frac{\sum_{v'=0}^{K-1} \sum_{u'=0}^{K-1} I(x+u', y+v') T(u', v')}{\sqrt{\sum_{v'=0}^{K-1} \sum_{u'=0}^{K-1} I(x+u', y+v')^2 \times \sum_{v'=0}^{K-1} \sum_{u'=0}^{K-1} T(u', v')^2}} \quad (6)$$

式(7)に示すように、切り取った矩形の原点である左上の座標値 (u'_0, v'_0) を (u_{match}, v_{match}) に加えることで、カメラ γ' におけるマッチングにより推定された注視点の位置 (u, v) (図 5.1 および 5.2 における緑丸)を求める。

$$u = u_{match} + u'_0, v = v_{match} + v'_0 \quad (7)$$

5. 処理結果

本実験では、データハンドリングおよび映像のズレ補正を行う計算機として、OS : Windows 8.1 Pro, CPU : Intel Core i7-4770 3.40GHz, GPU : Intel HD Graphics 4600, メモリ : 8.00GB[RAM]を搭載したデスクトップ PC を用いる。

5.1. データハンドリングに関して

撮影実験を行い、撮影データを提案するデータハンドリング手法を用いて計算機に取り込んだ。使用したカメラは GoPro 社の GoPro HERO4 ブラックである。カメラの解像度とフレームレートは 3840[画素]×2160[画素], 30[fps]とした。およそ 1 時間の撮影実験を 25 回に分割し撮影したところ、カメラ 1 台あたりのデータ量は約 8GB であった。カメラは 20 台使用した。適用した結果、親フォルダ内にテイク 1 からテイク 25 までのテイク分のフォルダが作成され、各フォルダ内には、図 6 に示すように、[カメラ番号.MP4]となるファイルがカメラ台数分作成された。その際、カメラ 1 台あたりに要した処理時間は約 1 分であった。

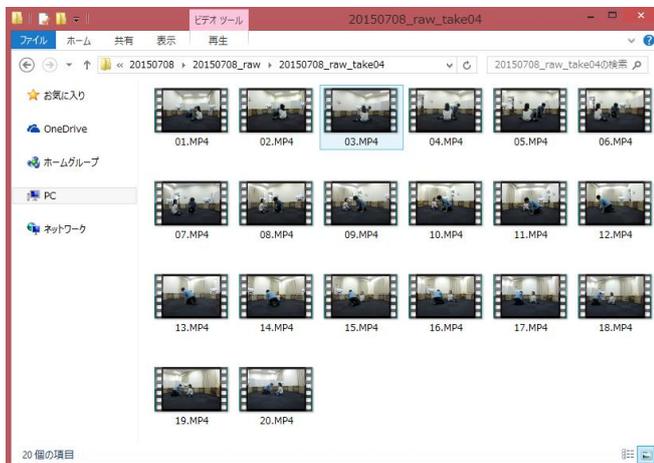


図 6. 撮影データを各テイクフォルダに保存

5.2. カメラズレの影響軽減処理

撮影初期時に設置するテクスチャの多い物体を図 7 のようにメディアサークルの内外に配置した。メディアサークルは円形状に組み立てた。また、SfM ベースのカメラキャリブレーションには一般に公開されている VisualSfM[7]を用いた。VisualSfM によって推定されたカメラキャリブレーションパラメータをもとに射影変換行列 P を生成し、本閲覧システムを実行した。

撮影初期の多視点映像に対しては注視点が全てのカメラ画像に対し同一箇所に集まり、推定されたカメラ位置・姿勢が正しいことが確認できた。

一方、撮影後半の多視点映像に対しては、図 8 に示すように、カメラ 1,2 において注視点が実際の位置より左下に現れた。これは撮影途中にカメラ 1,2 の位置がズレたためである。そこで 4 節で述べた映像補正処理を適用した。注視点を被写体の無地のシャツに設定した。カメラ 2 に対しテンプレートマッチングを行った様子を図 9 に示す。今回は動いていないカメラとしてカメラ 3 の注視点周りの領域をテンプレート画像として使用した。推定前のずれた注視点を赤丸、マッチングにより推定された注視点を緑丸で示す。図 9 から正しい注視点位置が推定できたことが分かる。得られた行列 D を適用した結果、正しい注視点位置に移動していることが確認できた。同様の処理をカメラ 1 にも適用することで、視点切り替え時の映像の不連続性を低減し、映像の円滑化に成功した。



図 7. 撮影実験開始時の環境

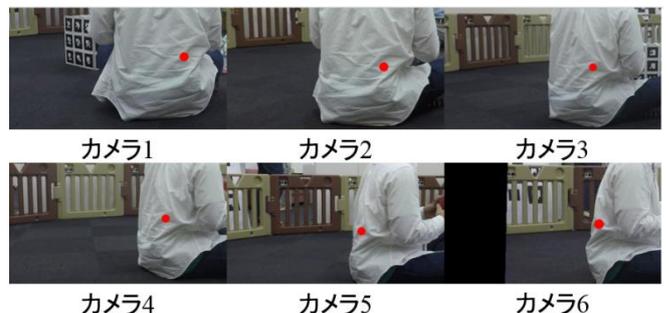


図 8. 注視点の位置 : 赤丸

6. おわりに

本稿では、多視点映像を **Bullet-Time** の生成に必要な情報と共に計算機に取り込むデータハンドリング手法と、被写体の衝突によるカメラ位置ズレが発生した場合に、再カメラキャリブレーションを行わずに、視点切り替え映像の品質を維持する補正手法を提案した。実証実験を実施することで、提案したデータハンドリング手法および映像補正処理の有効性を確認した。

本研究の一部は、科学技術振興機構(JST)の戦略的創造研究推進事業(CREST)における研究領域「人間と調和した創造的協働を実現する知的情報処理システムの構築」の研究課題「ソーシャル・イメージング：創造的活動促進と社会性形成支援」により行ったものである。

文 献

- [1] 坂本竜基, 陳鼎, “解像度を維持しつつ自然な **Bullet Time** を実現する射影変換の検討,” 情報処理学会インタラクション 2014, pp.65-70, February.2014.
- [2] 志田全弘, 明智奈央, 北原格, 大田友一, 松田壮一郎, 山本淳一, 鈴木健嗣, “簡単に組み立て可能な多視点映像の撮影および閲覧方式,” 電子情報通信学会 CNR/PRMU 研究会, 信学技報, vol. 114, no. 455, CNR2014-57, pp. 143-148, February.2015.
- [3] N. Akechi, I. Kitahara, R. Sakamoto, Y. Ohta, “Multi-Resolution Bullet-Time Effect,” ACM SIGGRAPH-Asia, December.2014.
- [4] N. Snavely, S.M. Seitz, R. Szeliski, “Photo tourism: Exploring photo collections in 3D,” ACM Transactions on Graphics (SIGGRAPH Proceedings), 25(3), July.2006.
- [5] O. Ozyesil, A. Singer, “Robust Camera Location Estimation by Convex Programming,” CVPR, June.2015.
- [6] Red Giant, “ビデオと別録りのオーディオが、ワンクリックで自動同期！パラ収録した複数の画と音を自動シンクロする映像音声同期ソフト,” http://www.flashbackj.com/red_giant/pluraleyes/, access:2015/9/6
- [7] C. Wu, “Towards Linear-time Incremental Structure from Motion,” 3D Vision, June.2013

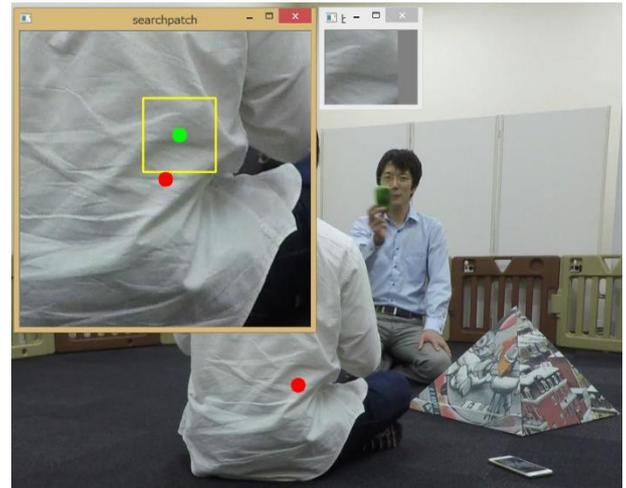


図 9. 提案するテンプレートマッチングによって推定された注視点位置(緑丸), 推定前のズレた注視点(赤丸), マッチング領域(黄色矩形)