Vol.2016-CG-162 No.6
2016/2/8

実写映像から推定した空間情報に基づくCGレンダリングに関する基礎研究 －その１：実写映像から３次元情報を抽出－

松本 颯生[1]　安藤 大地[1]　笠原 信一[1]

概要：実写画像へのCG合成の手法技術は進歩している一方で、実写動画へのCG合成技術は十分でなく, クオリティの高い合成には労力時間費用がかかる. そこで, 本研究は, 実写動画に違和感なくCGを合成するために, 自動的に実写動画から空間情報を読み取り, その情報を反映させたCGレンダリングシステムの開発を目指す. その第一段階として, 本稿では実写映像から３次元情報を抽出する手法を提案する.

キーワード：実写映像,CG合成, レンダリング,3次元情報推定

Fundamental Research for CG Rendering Based on Space Information Estimated from Motion Picture of Real Scene -Part1: Extracting 3D Information from Motion Picture -

Satsuki Matsumoto[1]　Daichi Ando[1]　Shinichi Kasahara[1]

Abstract: Technology for CG composition to a picture of real scene is progressing, but technology of CG composition to motion picture of real scene isn't enough, and it requires effort time-cost in high quality composition. So this research aims development of the CG rendering system recognizing spatial information from motion picture of real scene automatically and reflecting the information, for natural matching of CG objects in real scene. As the first step, this paper proposes improvement of the method to extract three-dimensional information from motion picture of real scene.

Keywords: Motion picture of real scene, CG composition, Rendering, Extracting 3D Information

1. はじめに

今日CG技術は大きな進歩をしている. 広告業界におけるポスターやCMなどから, 映画やテレビ業界におけるVFXにわたるまで、静止画から動画のさまざまな作品に利用されている.

その中でも実写へのCG合成は, より違和感なく合成されるようになっている. 特に静止画に関しては研究が進んでおり, 後述するKarschら[1]による研究により違和感の少ないレンダリング合成がほぼ人の労力なしに可能となっている.

しかし, 映像へのCG合成技術はまだ十分ではない. 映像の場合, 撮影された映像に合わせて動き明るさ影映り込みなどを合成するCGに反映させなければならない. 撮影された映像のカメラの動きを自動的に読み取り, それをCGの動きに反映させるものはすでにあるが, 明るさなどの調節まで含めてマッチングするシステムで実用化までできているものはない. そのため, クオリティの高いCG映像作品を作るには多くの労力と時間がかかってしまう.

そこで, ほとんど人の手間をかけることなく自動的に実写映像から３次元情報を読み取り, その情報を反映させた

[1] 首都大学東京　システムデザイン学部
Department of System Design
Tokyo Metropolitan University

CG レンダリングのシステムを開発を最終ゴールに設定して研究を進めている.

2. 研究の目的

実写映像に CG 物体を合成する際, 位置合わせと色合わせの 2 段階が必要である. 位置合わせとは CG 物体を配置したい場所にあるように見える位置に合わせることであり, 色合わせとは配置された CG 物体の明るさ・色合い・影を周りの状態に合わせて設定することである. 位置合わせの自動化に関する研究は多く行われており, マッチムーブ [2] などがこれにあたる. それに対し色合わせに関しては自動化の研究はあまり進んでいない. その手順として,CG 物体がどんな空間のどの場所に配置されているかという情報をもとに, 撮影された空間の 3 次元復元を行い, 復元された 3 次元モデル上にイメージベースドレンダリングにより光源の設定やテクスチャなどの追加を行って CG 物体のレンダリングを行う.

本稿ではこの手順の第一段階としてカメラの位置情報や写っている物体の 3 次元位置情報の抽出を行う. この手法を行う際に必要となる 3 次元情報は, 撮影された空間にある物体の大まかな形がわかればよく, スケールは実際の大きさではなく相対的なスケールで求められれば良い. 必要最低限の 3 次元情報のみを抽出することで計算量を減らすことが可能となる. しかし抽出される情報が少ない分, 復元の際などにズレが生まれやすいので, 抽出する必要最低限の 3 次元情報は抜けがないものが求められる. そこで本稿では, 既存の研究の改良を行うことで, より精度の高い必要最低限の 3 次元情報を実写映像から抽出することを目的とする.

3. 先行研究

Karsch ら [1] は実写の静止画に対して CG で作られた物体をほぼ違和感のなくレンダリングできる研究を発表している. その手順は写真に写っている空間のジオメトリや物体の形などを推定し, その後写真に写った影や窓から入り込む太陽光などから照明の位置を推定し, 以上の情報を参考に明るさや影, 色合いなどを計算するというものである.

この技術は精度が高く, ほとんど違和感のないレンダリングを実現している. しかし, この研究を動画に応用しようと考えると, 各フレームごとにこの動作をしなくてはならないため計算量が膨大となる, 各フレームで推定するためズレが生じてしまう恐れがあるなどの問題がある.

一方, 実写動画へのレンダリング技術としてはマッチムーブ [2] という技術が有名である. これは撮影された映像からカメラの動きを推定するもので, すでに専門業界でも使用されるほど確立されている.

この技術では, 動画の各フレームに写っている同じ点を追いかけ, 動きを見ることでカメラの動きを推定する. そして CG 物体をレンダリングする際のカメラの動きを, 推定されたカメラの動きに合わせることで, 映像中における CG 物体の位置を合わせることができる.

位置合わせを自動で行えるようになったことで CG 合成した映像の製作の負担が大きく軽減された. しかし,CG をレンダリングする際に明るさ, 影, 色合いを調整することも位置合わせと同じかそれ以上の負担がある.

一方, 動画から写っているものの 3 次元情報を抽出する技術としては, 複眼カメラによって撮られた映像から 3 次元情報を抽出する技術は様々に行われているが, それでは映像製作の現場などでの活用は難しい. しかし, 関ら [3] によって単眼カメラで撮影された動画からの 3 次元情報抽出の研究が行われている. これは, ある特徴的な点 1 点が各フレームでどこに写っているかを追いかけることでカメラの動きを推定し, 同じものが映像の各フレームでどこに写っているかという位置関係を対応させ, それをもとに三角測量の原理によって写っているものの 3 次元位置を求めるというものである. これにより, 一般的に使われている単眼カメラで撮影された映像からでも写っている物体の 3 次元形状の復元を可能としている. しかしこれは写っている物体の細部まで細かく復元しているために計算量が膨大となってしまっている. 一つの物体対してのみを検討する場合にはとても有効であるが, 空間全体を把握するのには難しい.

3.1 動画からの 3 次元情報抽出の手法

関ら [3] は以下の方法で動画の 3 次元情報を抽出する方法を提案している. 撮影された映像の各フレームで特徴点を抽出する. そしてそれぞれのフレームで特徴点を対応させる. 対応した特徴点が各フレームでどの位置に写っているかの座標情報をもとにエピポーラ幾何 [*1] の考え方で 3 次元座標を求める. この時カメラの動きも特徴点の一つが写っている場所の動きにより求められる.

エピポーラ幾何とは, 視点とフレーム上の投影点を結んだ直線上に見ている点もあることから, 同じ点を投影していた場合, 2 つのフレームにそれぞれ写る点とそれぞれの視点を結んだ 2 つの直線の交点が見ている点の 3 次元座標となる考え方である (図 1).

3.2 特徴点の抽出と対応の手法

特徴点の抽出と対応を行う手法として,SIFT アルゴリズム [*2] がある. SIFT は特徴点抽出と特徴量の記述, 対応を行う. その処理は以下の通りである.

(1) 対象とする 1 枚の画像にガウシアンフィルター [*3] によってぼかし加減を段階的に変化させ, 画像群を作る.2

[*1] エピポーラ幾何:2 つの異なる視点の画像から 3 次元情報を再構築する幾何.
[*2] SIFT アルゴリズム:入力された画像から特徴点の検出と特徴量の記述を行うアルゴリズム.
[*3] ガウシアンフィルター:画像の平滑化を行うフィルター.

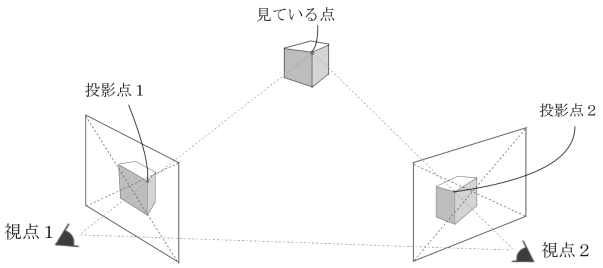

図 1 エピポーラ幾何
Fig. 1 Epipolar geometry

倍にぼかすと解像度が 1/2 の画像と同じになるので, 段階的にぼかし具合を変えるとスケールが徐々に小さくなる画像群を作れる.

(2) このようにして作られた各解像度の画像の差分をそれぞれ出力することで DoG 画像 [*4] を作る.

(3) 注目画素をその周囲画素と比べ,DoG 画像間の解像度方向と画像の XY 方向の 3 次元方向で極地を探す.3 次元方向全てが一番大きく変化している座標とその時のスケールを決定する. 極地となった点が特徴点として抽出される.

(4) 次に (3) で決定されたスケールにおいて, 特徴点の周囲領域で輝度の大きい方から小さい方への勾配情報をオリエンテーションとして記述する.

(5) 特徴点を中心として特徴点のオリエンテーションの大きさを半径とした周辺領域を 4 × 4 の 16 ブロックに分割し, ブロックごとに 8 方向のオリエンテーションを作成する. これを特徴量の情報として記述する.

(6) 抽出された特徴点を, 特徴量を元に同じと思われる特徴点同士を対応づける.

3.3 コーナー点抽出の手法

3.3.1 FAST

注目ピクセルとその周囲のピクセルの輝度を比べ,Brighte, Similar, Darker の 3 グループに分ける (図 2).

$$Brighter : Ip + th \leq Ix \quad (1)$$

$$Similar : Ip - th < Ix < Ip + th \quad (2)$$

$$Darker : Ix \leq Ip - th \quad (3)$$

Ip=注目点の輝度
Ix =周囲の点の輝度 (x は周囲のピクセルの番号)
th =閾値

そして, 周囲のピクセル数 16 の内の 9 つ以上が連なって Brighter もしくは Darker である場合コーナー点とみなすという考え方である (図 3).

[*4] DoG 画像:平滑化された 2 枚の画像の差分を出力した画像.

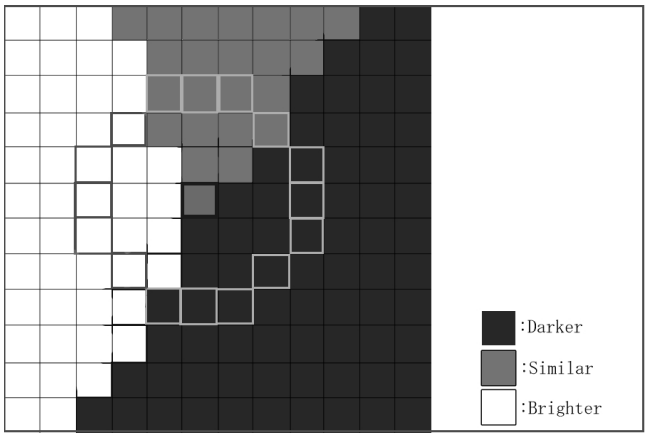

図 2 注目画像の周囲ピクセルのグループ分け
Fig. 2 Grouping surrounding pixel of attention picture.

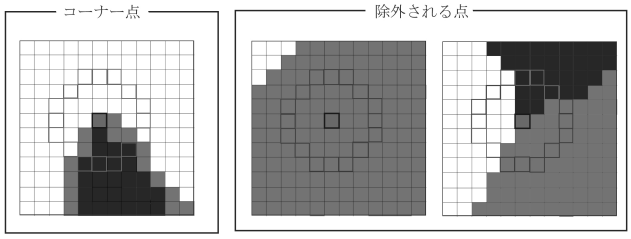

図 3 コーナー点とみなされるものと除外されるもの
Fig. 3 Assumed as corner and deselected.

3.3.2 FAST machine learning

あらかじめ FAST のアルゴリズムによってコーナー, 非コーナーとされた大量の画素からを機械学習により決定木 [*5] を学習する. その決定木を利用してコーナー点を検出する. それにより速度の向上が望める.

3.3.3 Cascaded FAST

注目ピクセルの周囲 16 個だけでなく,20 個と 12 個を加えた 3 重の周囲ピクセルを FAST のアルゴリズムで判定する. また,3 重の周囲ピクセルのオリエンテーションの類似性 によってよりコーナー点らしい点だけを選択する.

4. 改善手法の提案

4.1 コーナー点のみを利用した 3 次元情報抽出

関らの手法では特徴点すべての 3 次元情報を抽出していたため計算量が多かった. また,3 次元モデルを作り, 形を把握することを目的としているため,3 次元情報を抽出した特徴点のすべてを 3 次元上に点を打つことで映されているものを再現しているが, 点の集合であるため, どこが面か境界線かなどがわからない. 本研究では 3 次元モデルを再現したのちそのモデルを基に影や照明の計算を行いたい. そのため面や線が必要となってくる. そこで, 特徴点すべてではなくコーナー点のみの 3 次元情報を抽出し, それを結ぶことで面や線の再現も行う.

[*5] 決定木:データの中から注目する領域をみつける方法.

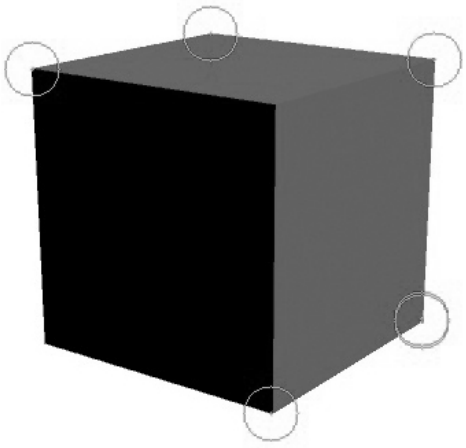

図 4 FAST によるコーナー点検出
Fig. 4 Detecting corners by FAST.

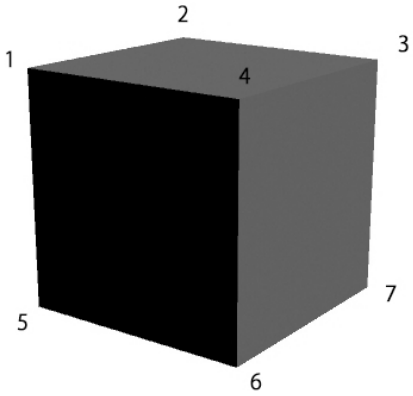

図 5 コーナー点として検出されるべき点
Fig. 5 Points which should be detected as corner.

4.2 FAST の改善手法

前章で紹介した FAST アルゴリズムでは, コーナー点を十分に抽出しない (図 4).

16 分の 9 という閾値では, 取り逃がすことがあると考えられる.16 分の 9 という閾値は直線上の点は省かれ, コーナーは含まれる値であるが, それは周囲ピクセルが Similar,Darker,Brighter の 3 種類によって 3 分の 1 ずつ構成されていた場合コーナーとみなさなくなる. つまり図 5 の点 4 や影ができれば点 6 などもコーナー点とみなさない.

FAST において閾値が大きすぎたために十分にコーナー点を抽出できなかったと考えられる.FAST は周囲を 3 種類のグループに分けるので, それぞれのグループが 3 分の 1 ずつを取っていた場合 (図 4 の点 4 のような場合) コーナーとして取らなくなる. そこで, 周囲 16 ピクセルの 3 分の 1(5.333…ピクセル) を取るグループも含ませるため, 16 ピクセル中 6 ピクセル, つまり閾値を 8 分の 3 まで小さくする. しかし, 閾値を小さくするとコーナー点以外にも直線上の点や 1 つのコーナー点周りで多くの点を取ってしまう.

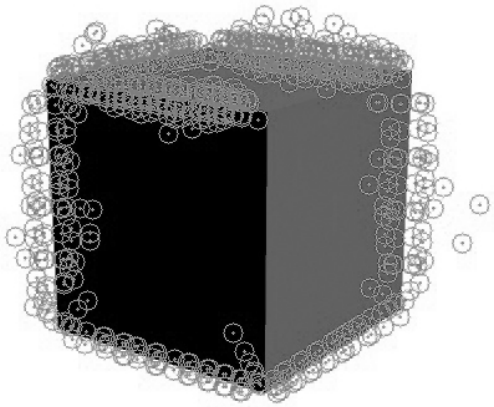

図 6 SIFT による特徴点検出
Fig. 6 Detecting feature points by SIFT.

そこでオリエンテーションによる選別により, 余分な点を切り捨て, 実際のコーナーに一番近い点だけを残すようにする.

また,SIFT によって検出される特徴点にはコーナー点が含まれる (図 6) ので, この特徴点に関してだけコーナー点か非コーナー点かの評価を行う. また周囲のピクセルを円ではなく正方形でとる. この 2 つによりコーナー点の判別で計算量が増えてしまう分, 他の計算を簡略化する. FAST の考え方と上記の閾値やオリエンテーションによる選別をもとに改善手法を提案する. この手法の手順は以下の通りである.

(1) SIFT によって検出された特徴点を注目点とし, 注目点の周囲ピクセルを FAST と同じく 3 つのグループに分ける.
(2) コーナー点とみなす際の閾値を 8 分の 3 とし, 周囲のピクセル数の 8 分の 3 以上が連なって Brighter もしくは Darker の場合とコーナー点の候補とする (図 7).
(3) コーナー点とみなされた場合,8 分の 3 以上 Brighter もしくは Darker が続く箇所には, 注目点からその幅の真ん中へのベクトルを方向とし, Brighter もしくは Darker が続く個数を大きさとするオリエンテーションをつける (図 8). 一つの点に二つのオリエンテーションがあった場合は大きさの大きい方をメインのオリエンテーションとする.
(4) (2) においてコーナー点の候補とて挙がられた点の中で距離が近く, オリエンテーションの方向が同じものはオリエンテーションの大きさが大きい方だけ残す. 実際のコーナーにより近い点を取ることができる (図 9).
(5) また, 同じ方向のオリエンテーションが 3 以上ある場合は直線上の点とみなして排除する.

以上の実行結果を図 10 に示す. 点 1,2,10 など同じコーナー点周りで取ってしまっているが, これは距離が近いものは

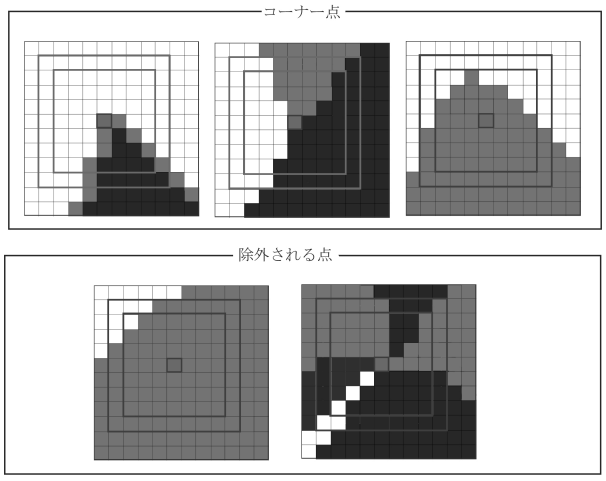

図 7 コーナー点とみなされるものと除外されるもの
Fig. 7 Assumed as corner and deselected.

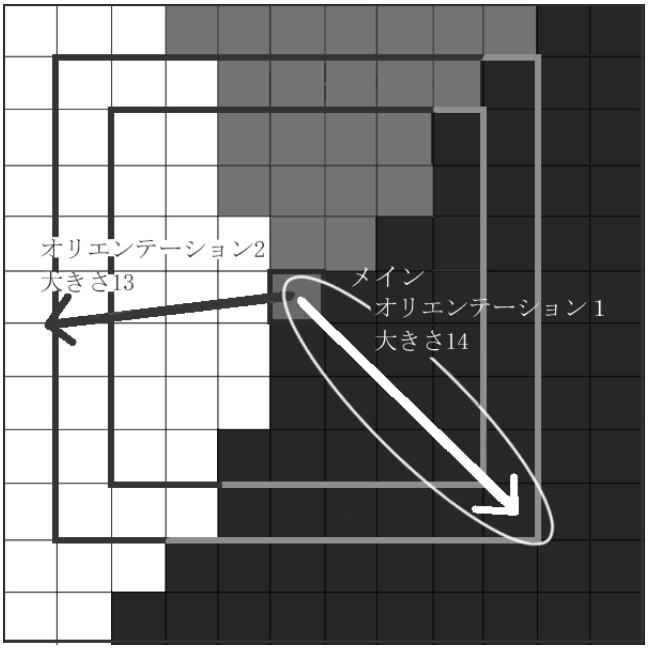

図 8 オリエンテーションのつけ方
Fig. 8 Configuring orientations.

まとめることで, 一つに絞ることができる.

4.3 3 次元座標の推定

本研究では, 実写映像を用いるため映像をフレームごとに分割し, 前章の提案手法を用いて全てのフレームでコーナー点の抽出を行い,SIFT の特徴量をもとに隣り合うフレームと対応させていく. そしていくらか離れたフレームどうしで, 関らの方法に沿って,3 次元座標を求める計算を行う. 本研究ではフレームに写るコーナー点の 2 次元座標などの誤差を考慮して, 交点ではなく最近点を求める. 求められた最近点 2 つの真ん中の点を見ている点の 3 次元座標とする.

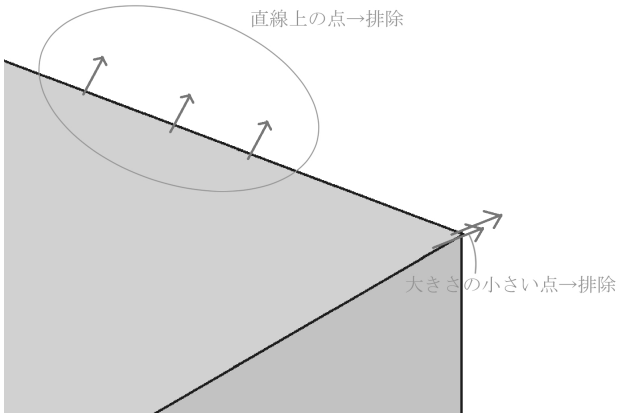

図 9 オリエンテーションによる選別
Fig. 9 Sorting out by orientations.

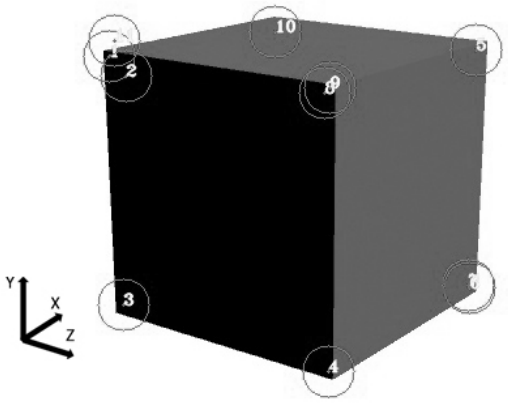

図 10 コーナー点推定結果
Fig. 10 Result of estimate corners.

3D モデラーで作成した立方体を周囲から写す動画 (60 フレーム) を入力し, 上記の流れを実行した.1 枚目のフレームと 60 枚目のフレームで 3 次元座標を計算した. 立方体は 3D モデラーで作成したので, 各頂点の座標がわかる. 代表的な 3 点に対して表 1〜3 に 3D モデラー上での実際の座標と提案手法によって推定された 3 次元座標の比較を示す. ただし, 点の番号は図 10 に示したもので,X 座標を奥行き,Y 座標を高さ,Z 座標を横方向としている.

X 座標のズレが大きい原因はカメラの Y 座標や Z 座標の向きの変化が少なかったためである (表 4). 計算に使用するフレーム間の枚数を増やし, 角度をつけた 2 方向からの画像を使用することでズレを小さくできると考える.

表 1 点 2
Table 1 Point 2.

	X	Y	Z
実際の座標 [mm]	15.4	5.8	-214.6
求められた座標 [mm]	27.5	1.8	-224.0

表 2 点 6
Table 2 Point 6.

	X	Y	Z
実際の座標 [mm]	-38.5	-46.4	-16.14
求められた座標 [mm]	-29.5	-49.8	-168.8

表 3 点 7
Table 3 Point 7.

	X	Y	Z
実際の座標 [mm]	14.6	-47.2	-161.3
求められた座標 [mm]	26.6	-51.2	-169.7

表 4 カメラの座標
Table 4 Coordinates of camera.

	X	Y	Z
1 フレーム目 [mm]	-200	-50	-50
60 フレーム目 [mm]	-155	-50	-36

5. 今後の研究の進め方

前章の課題を解決した上で, 推定された 3 次元モデルに, 今後イメージベースドレンダリングの技術を利用してテクスチャや照明の情報を追加することで, 撮影された空間を擬似的に再現することを行う. その空間上に CG 物体を配置しレンダリングを行った結果を, 実写映像に合成することでその空間に溶け込んだ違和感のない合成を目指す.

参考文献

[1] Kevin Karsch, Varsha Hedau, David Forsyth, DerekHoiem “Rendering synthetic objects into legacy photographs” ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia),30(6),2011
[2] マッチムーブ: 動画編集技術「マッチムーブ」 http://news.mynavi.jp/series/computer_vision/002/
[3] 関 晃仁, オリバー ウッドフォード, ”単眼カメラの動画像を用いたリアルタイムで緻密な 3 次元再構成技術 ” 東芝レビュー 5 月号,VOL.68 NO.5,2013
[4] E.Rosten and T.Drummond, ”Fusing Points and Lines for High Performance Tracking ”Computer Vision,ICCV 2005
[5] E.Rosten and T.Drummond, ”Machine Learning for High-Speed Corner Detection ” Computer Vision,ECCV 2006
[6] 長谷川昂宏, ” Cascaded FAST による特徴点検出 ” 電子情報通信学会論文誌, Vol.J98-D, No.4, pp.560-570, 2015
[7] 竹田遼, 内田祐介, 酒澤茂之, 半谷精一郎, ” 複数の決定木を用いた高速な FAST 特徴点検出 ” 第 74 回全国大会講演論文集,2012(1),507-508 ,2012
[8] 藤吉弘亘, “ Gradient ベースの特徴抽出-SIFT と HOG-“ 電子情報通信学会技術研究報告. PRMU, パターン認識・メディア理解 107(206), 211-224, 2007
[9] 古畑俊一郎, 亀田能成, 大田友一, ”領域を限定した SIFT 特徴の抽出 ” Proceedings of Meeting on Image Recognition and Understanding (MIRU2008), pp.1330-1335, 2008
[10] 西村孝, 藤吉弘亘, “ 画像局所特徴による対応点マッチングとその応用に関する研究 ” 中部大学 工学研究科 情報工学専攻 修士論文 (2009)