

GPS シグナルを利用した 疎な Google Street View 画像列の時間軸補間

伊藤 晟^{1,a)} 鮫島 正樹^{2,b)} 前川 卓也^{2,c)} 松下 康之^{2,d)}

概要：Google Street View は道路に沿って一定間隔で撮影された風景画像を世界規模で提供しており、その画像列を用いたタイムラプス動画の生成には一定の興味がある。しかし、Google Street View で提供されている画像列は疎であるため、直接的に動画を生成すると、隣接フレーム間の差が大きい動画が生成されてしまう。そのため、本稿では疎な画像列を入力とし、時間軸補間により密な中間画像列を生成する方法について検討する。画像列の時間軸補間において対応点探索が重要な問題となるが、既存手法では一般に、見えの差が大きい画像間の対応点を見つけることは困難であるという問題があった。提案手法では各画像に関連付けられた GPS シグナルを利用して予測できる基礎行列を手掛かりに、対応点の探索範囲を制限することで高精度な対応点探索を実現する。提案手法を用いて得られる対応点を利用することで、疎な画像列の時間軸補間の精度を向上させた。

ITO AKIRA^{1,a)} MASAKI SAMEJIMA^{2,b)} TAKUYA MAEKAWA^{2,c)} YASUYUKI MATSUSHITA^{2,d)}

1. はじめに

風景などを撮影した画像を時間的、空間的に並べて作成されるハイパーラプス動画は撮影シーンの変化を把握するうえで有用である。現在地図サービスとして提供されている Google Street View や Bing Maps の市街地画像列では、道路に沿って一定間隔で撮影された世界中の風景画像を提供しており、それらの画像を用いて全世界の道路を旅しているようなハイパーラプス動画を作成することができる。Derek [1] は Google Street View の画像を用いてハイパーラプス動画を生成するアプリケーションを公開している。しかし、これらの画像列は一定間隔で撮影された疎な画像列であるため（例えば Google Street View では約 10m 間隔である）、単純な画像連結のみでは隣接フレーム間の見えの差が大きいため滑らかなハイパーラプス動画とならない。そこで本研究では、このような疎な画像列を補間することにより、疎な画像列から密な画像列を生成し、滑ら

かなハイパーラプス動画を作成することを目指す。画像列の時間軸補間は、隣接フレームの対応点を探索し、対応点を用いて補間画像を作成する方法が一般的である。実際の風景と近い画像を時間軸補間するためには、対応点探索を高精度に行うことが重要である。しかし本研究で扱う疎な画像列では隣接フレーム間の見えの差が大きく、一般的な対応点探索の方法では時間軸補間に十分な精度を得ることができない。

そこで、提案手法では各画像に関連付けられている GPS シグナルを利用して画像の各特徴点の対応先を予測する。この予測される対応点の範囲に限定して対応点探索を行うことで正しい対応点を推定し、疎な画像列の補間精度を向上させる。

提案手法では、各特徴点の対応先を予測するために、GPS シグナルに基づいて補間対象の 2 枚の画像間の基礎行列を求める。求めた基礎行列をもとに、対応点の探索範囲を限定することにより、効率的かつロバストな推定を試みる。さらに、画像の領域分割を用いて外れ値に相当する対応点を除去することで mismatches を低減させる。このようにして得られた対応点を利用し、パラメトリックな画像ワーピングにより補間画像を生成する。

¹ 大阪大学工学部

School of Engineering, Osaka University

² 大阪大学大学院情報科学研究科

Graduate School of Information Science and Technology, Osaka University

a) ito.akira@ist.osaka-u.ac.jp

b) samejima@ist.osaka-u.ac.jp

c) maekawa@ist.osaka-u.ac.jp

d) yasumat@ist.osaka-u.ac.jp

2. 関連研究

画像列もしくは動画をフレーム補間する研究は過去に多く行われている。補間フレームの生成のためには与えられたフレーム間の対応点探索と画像ワーピングが重要な技術となる。

2.1 対応点探索

画像間の対応点探索についての研究は過去に多く行われている。画像列が密な場合には、ブロックマッチング [2] やオプティカルフロー [3] などが用いられる。これらの手法は動画の隣接フレームにおいて見え方がほとんど変化しないという考えに基づいている。そのため画像列が疎な場合、これらの手法を直接的に適用することで対応点を探索することは困難である。

見えの差が大きい画像間の対応点探索については、局所特徴量を利用した対応点探索の手法が盛んに研究されている。例えば、Lowe [4] は画像の回転・スケール変化・照明変化等に頑健である SIFT 特徴量を用いることで見えの差が大きな画像間の対応点探索を可能にした。また、Bay ら [5] は高速な計算が可能な SURF 特徴量を提案している。

2.2 画像ワーピング

Choi [6] はブロックマッチングをベースに各ピクセルの動きを推定することで補間画像の作成を行っている。Seitz と Dyer [7] は対応点から画像間の基礎行列を求めることにより同じものを撮影した 2 枚の画像を用いて、2 枚の画像が撮影されたカメラ位置の間で撮影される画像を生成している。これは 2 枚の画像に同じ被写体が写っているという前提に基づいて行っているため、本稿で対象としている多くの被写体がどちらか一方の画像にしか写っていない画像に適用することは難しい。また、Gurdan ら [8] は SIFT 特徴量に加えて Harris と Stephens [9] の手法を用いることで対応点探索の精度を上げ、Seitz と Dyer の手法 [10] を応用することで補間画像を作成している。この手法は対応点を用いて三角形のメッシュを作りメッシュごとにワーピングさせているが、本稿で扱う画像では、様々なオブジェクトが重なり合っている上に、細かいオブジェクトの対応を取ることが難しいため、この手法を直接用いるのは困難であると考えられる。また、Yang ら [11] は人の顔が映されている画像から別の顔が映されている画像への変化が自然となるように、顔の構造について事前に学習を行うことで補間画像を生成する研究を行っている。入力画像列が密な場合はこれらの研究のように各ピクセルの動きを推定することでノンパラメトリックなワーピングが可能であるが、入力画像列が疎である場合は各ピクセルの動きベクトルを推定することが困難であるためパラメトリックモデルを用い

る必要がある。

パラメトリックなモデルには自由度が高い Thin-plate spline [12] や、よりリジッドな射影変形などがある。

3. 提案手法

提案手法の手順を図 1 に示す。まず GPS シグナルが関連付けられている画像列 (本研究では Google Street View を用いてカメラを進行方向に向けた時に得られる大きさ 640×640 pixels の画像を使用) から基礎行列を推測し、各特徴点の対応先を予測する。次に予測される対応点の範囲において対応点が満たすべき条件を考慮することで高精度な対応点探索を実現する。最後に、得られた対応点を利用して補間画像を作成する。

3.1 基礎行列の推定

まず、隣接フレーム間の対応点の手前の画像座標を (x_1, y_1) 、後の画像の座標を (x_2, y_2) とする。基礎行列 F は

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix}^T F \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = 0 \quad (1)$$

となるため、基礎行列が分かると特徴点の対応先の座標を線形式で予測することができる。また、対応点 (x_1, y_1) と (x_2, y_2) を結ぶ直線を各対応点について引いたときこれらの直線は一点で交わる。この交点を消失点と呼ぶことにする。基礎行列はカメラの外部パラメータと内部パラメータを用いて求められる。カメラの外部パラメータは、手前の画像を撮影したカメラから後の画像を撮影したカメラの回転行列 R 、並進ベクトル T で構成される。Google Street View では、GPS データから撮影したカメラの緯度経度及びカメラの向きを取得することができるため、回転行列と並進ベクトルは計算することができる。カメラの内部パラメータ A は画像列からの正確な値を求めることが難しいため、焦点距離を様々な値に変えて消失点を計算し、もっともらしい場所に消失点が出現する値とした。

カメラの外部パラメータおよび内部パラメータから基礎行列を求める。 T と $\begin{bmatrix} x_2 & y_2 \end{bmatrix} A^T$ と $RA^{-1} \begin{bmatrix} x_1 & y_1 \end{bmatrix}^T$ は図 2 のような関係となり、同一平面上に存在する。そこで T の歪対象行列を $\begin{bmatrix} T \\ \times \end{bmatrix}$ とすると式 (2) が成り立つ。式 (2) と式 (1) を用いて基礎行列を求めることができる。

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix}^T A^T \begin{bmatrix} T \\ \times \end{bmatrix} RA^{-1} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = 0 \quad (2)$$

$$F = A^T \begin{bmatrix} T \\ \times \end{bmatrix} RA^{-1}$$

基礎行列を推定することで各画像の消失点を推定することが可能である。

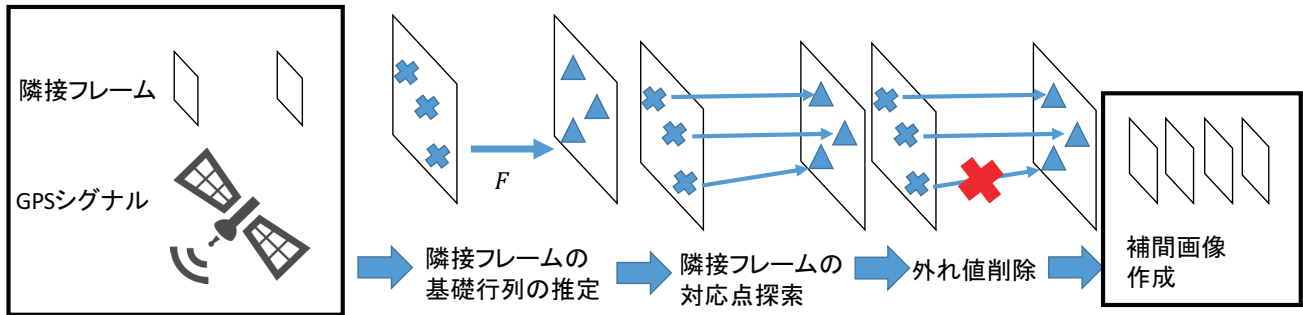


図 1: 提案手法の流れ

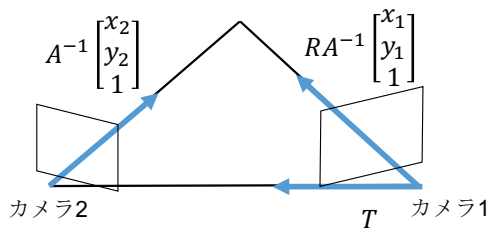


図 2: ベクトルの関係

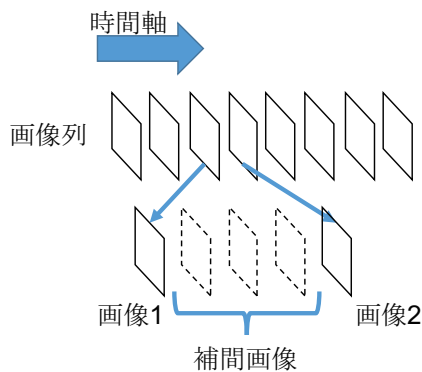


図 3: 画像の関係

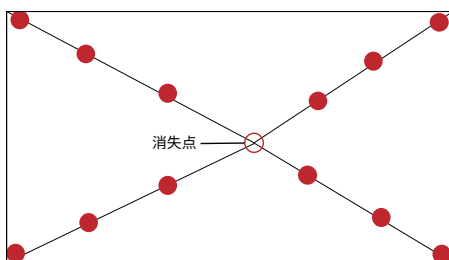


図 4: 画像を消失点を用いて4つの領域に分割する。消失点以外の点はホモグラフィ行列を求める際に境界線上の点として使用。

3.2 対応点探索

図 3 に示すように補間したいフレームの直前の画像を画像 1、直後の画像を画像 2 とする。まず各画像から SURF 特徴量 [5] を用いて特徴点を検出し、画像 1 の各特徴点の

対応先を画像 2 から探索する。画像 1 で遠くに撮影される物体は、前方に進んだ画像 2 では手前に撮影される。すなわち画像 1 の特徴点に対応する画像 2 の特徴点は、消失点からみて画像 1 の特徴点より離れた位置に存在すると予測できる。3.1 節で求めた基礎行列が誤差を含む可能性があるため、図 5 に示すように画像 1 の特徴点の座標と消失点を結ぶ直線から角度 θ だけ広げた領域から対応点を探索する。続いて画像 2 の各特徴点の対応先を画像 1 から探索する。画像 1 は画像 2 より手前で撮影されているため、画像 2 の特徴点に対応する画像 1 の特徴点は、消失点と画像 2 の特徴点の座標との間に位置すると予測される。そこで図 6 に示すように探索範囲を中心角 θ の扇形の内側に制限する。前方、後方の両方から探索した対応点のうち、両方で一致した対応を得られるもののみを対応点として採用する。

続いて得られた対応点の外れ値を検出し削除することを考える。まず、2 枚の画像の対応点の画素値はほぼ同じであると考えられるため、前方、後方から求めた対応点の画素値の差が一定以上であれば、外れ値とみなして対応点から削除する。さらに、座標が近い特徴点は、対応先の座標も近いと考えられるため、周囲の特徴点と比較して対応先の座標が大きく異なる特徴点を削除する。これを実現するために画像 1 に SLIC 法 [13] を適用し、画像 1 を Superpixel に分割する。SLIC 法とは画像を Lab 色空間に変換し、色の類似度や位置関係を用いて Superpixel とよばれるクラスターに分割する手法である。各 Superpixel において存在する対応点の移動距離を d_i ($i = 1, 2, \dots, n$) とし、式 (3) と式 (4) より平均 μ と分散 σ^2 を計算し、分散が τ 以上の場合外れ値があるとみなし、移動距離が平均から最も離れている対応点を削除し再び分散を計算する。これを分散がしきい値 τ 以下になるまで繰り返す。

$$\mu = \frac{\sum_{i=1}^n d_i}{n} \quad (3)$$

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (\mu - d_i)^2 \quad (4)$$

3.3 補間画像作成

画像1の時間を0, 画像2の時間を1とした時に補間したい画像の時間を α ($0 < \alpha < 1$)とする. まず, 対応点を線形補間により補間画像に移す. 移した対応点の座標 (x_i, y_i) は

$$x_i = (1 - \alpha)x_1 + \alpha x_2$$

$$y_i = (1 - \alpha)y_1 + \alpha y_2$$

となる. この点を基準に画像1を変形し, 補間画像を作成する. 補間画像を生成するにあたり, 画像ワーピングのモデルにはいくつかの候補が考えられる. 例えば, 単純なホモグラフィ変換に基づくもの, あるいはより柔軟なワープを表現できる Thin-Plate Spline 法などがある. 単純なホモグラフィ変換では画像全体を一つの面として変形するが本稿で用いる画像は道路・建物・空が大きく映っており, これらは明らかに同一平面上に存在しないため, 直接使用することは困難である. また, Thin-plate spline 法は自由度が高く, 実験的に画像が不必要に歪んでしまうという課題があった. そこで本研究では, 消失点を用いて画像1を図4に示すように消失点から画像の4隅に境界線を引くことで道路・建物・空がそれぞれ大きく映っている4つの領域に分割する. 4分割した各領域に存在する対応点を利用し, 区分的に定義されるホモグラフィ行列を最小二乗法により計算する. ホモグラフィ行列とは平面を別の平面に変形させる行列であり, 変化前の座標を (x, y) , 変化後の座標を (x', y') とすると

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

と表現できる. 区分的に定義されるホモグラフィ行列を計算する際に, 境界線上の点は隣接する領域のどちらのホモグラフィ行列を用いて移動させても同じ位置に移動するように位置の誤差にペナルティとなる重み w をつけて境界線が目立たないようにする. 消失点に対して上の領域を領域1, 右の領域を領域2, 下の領域を領域3, 左の領域を領域4とする. 領域 m のホモグラフィ行列を $H^{(m)}$ とし,

$$H^{(m)} = \begin{bmatrix} h_{11}^{(m)} & h_{12}^{(m)} & h_{13}^{(m)} \\ h_{21}^{(m)} & h_{22}^{(m)} & h_{23}^{(m)} \\ h_{31}^{(m)} & h_{32}^{(m)} & h_{33}^{(m)} \end{bmatrix}$$

とする. 画像1の対応点の座標を (x, y) とし, その対応先の補間画像での座標を (x', y') とする. そして, 画像1の領域aと領域bの境界線上の任意の位置の点の座標を (x_{ab}, y_{ab}) とする. 4つの領域のホモグラフィ行列は式(5)を最小二乗法を用いて解くことで4つの領域のホモグラフィ行列を求める.

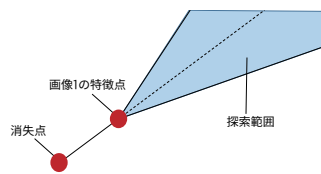


図 5: 画像2の探索範囲

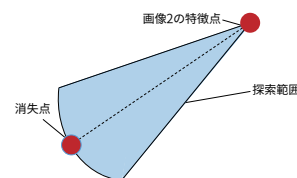


図 6: 画像1の探索範囲

$$\begin{bmatrix} x & y & 1 & 0 & 0 & \dots & \dots & \dots \\ 0 & 0 & 0 & x & y & 1 & 0 & \dots \\ & & & \vdots & \vdots & & & \\ wx_{12} & wy_{12} & w & -wx_{12} & -wy_{12} & -w & 0 & \dots \\ & & & \vdots & \vdots & & & \end{bmatrix} \begin{bmatrix} h_{11}^{(1)} \\ h_{12}^{(1)} \\ h_{13}^{(1)} \\ \vdots \\ h_{33}^{(1)} \\ h_{11}^{(2)} \\ h_{12}^{(2)} \\ \vdots \\ h_{33}^{(4)} \\ h_{32}^{(4)} \\ h_{33}^{(4)} \end{bmatrix} = \begin{bmatrix} 2x_1^1 \\ 2y_1^1 \\ 1 \\ 2x_2^1 \\ 2y_2^1 \\ 1 \\ \vdots \\ 2x_{n_1}^1 \\ 2y_{n_1}^1 \\ 1 \\ 2x_1^2 \\ \vdots \\ 2y_{n_4}^4 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (5)$$

4. 結果

実験では一貫して以下のパラメータを用いた. 3.2章で述べた探索範囲を限定する際に用いる θ を $\frac{1}{20}$ [rad], 分散に関するしきい値を $\tau = 5.0$, ホモグラフィ推定時に境界条件に使用する点数を12, その重み係数を $w = 30.0$ とした.

4.1 対応点探索

Google Street View で提供されている画像列に対して一般的な手法である SURF 特徴量のみを利用した手法と本稿で提案した手法によって得られた対応点と SLIC 法による領域分割の結果は直進時の画像列に対しては図7, 右に曲

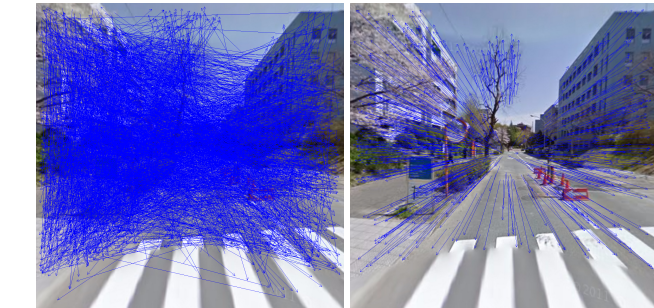


図 7: 直進時の画像列を用いた際、各対応点が画像 1(左上) から画像 2(上中央) でどのように移動しているかを青矢印で示している。左下が SURF 特徴量のみを使用したもの。右下が提案手法によるもの。右上の画像は画像 1 を SLIC 法により Superpixel に分解した結果

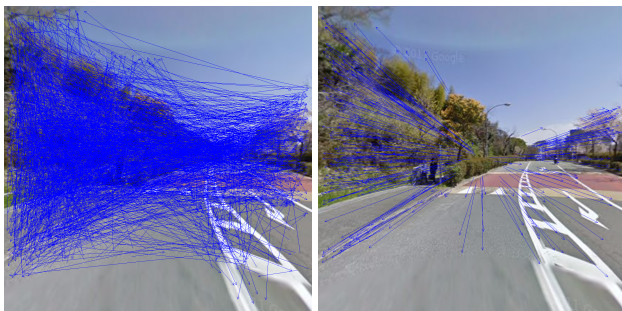


図 8: 右に曲がりながら進んだ時の画像列を用いた際、各対応点が画像 1(左上) から画像 2(上中央) でどのように移動しているかを青矢印で示している。左下が SURF 特徴量のみを使用したもの。右上の画像は画像 1 を SLIC 法により Superpixel に分解した結果

がりながら進んだ時の画像列に対しては図 8 のようになった。図 7 や図 8 の SURF のみを用いる手法では、対応先を示す矢印が様々な方向を向いており、誤った対応をしている点が多く存在するが、提案手法を用いて得られる対応点は消失点から放射線状に対応点が移動しており、対応点探索の精度が向上していることが確認できる。

4.2 補間画像作成

補間画像を作成し、画像枚数を 5 倍に増やすことを試みた。画像を 4 つに領域分割して区分的に定義されるホモグラフィ行列を計算する提案手法と、パラメトリックなモデルである Thin-plate spline 法を用いる手法とリジッドな手法である一つのホモグラフィ行列のみを計算する手法を用いて補間画像を生成した。それぞれの手法で生成した補間画像は図 9 や図 10 のようになった。図 9 では Thin-plate spline 法は自由度が高いため、道路左側にあるミラーなどが著しく曲がってしまい、本来ありえない形に変形してしまっている。これに対して、一つのホモグラフィ行列のみを計算する手法と提案手法を用いたものでは比較的歪みが少なく綺麗に補間されていることが確認出来る。しかし、一つのホモグラフィ行列のみを計算する手法では、道路左側の木やコーンが進み過ぎてしまっている。一方で、提案手法では図 9 の車道にあるコーンなどの移動量と奥にある建物の移動量とがほぼ同じであるかのように補間されてしまっているが進み過ぎを防ぐことができた。そのため、どの手法を用いてもどちらも補間画像と入力画像 2 を比べると差ができてはいるが、提案手法が入力画像と補間画像の差が最も小さく滑らかなハイパーラプス動画を生成できる。図 10 では、図 9 と同様に Thin-plate spline 法では建物など本来まっすぐなものが歪んでしまっているが、一つのホモグラフィ行列のみを計算する手法と提案手法ではまっすぐなまま補間が行えている。また、一つのホモグラフィのみを計算し補間した場合は画像平面を傾けたように補間されてしまうが、提案手法では他手法より綺麗に補間されている。

5. おわりに

本稿では疎な画像列を関連付けられている GPS シグナルを利用することで対応点探索の精度を向上させ、画像列の時間軸補間を試みた。提案手法では画像のみでは困難な見えの差が大きい画像間の対応点探索を、GPS シグナルを利用し消失点を求めることで対応点探索の精度を向上し、補間画像を生成した。補間画像を生成するためのホモグラフィ行列を求める際に、画像全体のホモグラフィのみを求め補間画像を生成すると、本来オブジェクトによって移動する量が異なるはずが、どのオブジェクトも同じ量しか動かない画像となってしまう。特に道路の移動量が小さくなり、補間画像と入力画像の差が大きくなるものとなる。そのため、画像を消失点を利用して分割し、それぞれの領域でホモグラフィ行列を計算することにより改善を試みた。補間画像は改善されたが依然として入力画像との差がある。提案手法では消失点を利用し画像を 4 分割する際に、消失点から 4 隅に境界線を引くという単純な方法で行っているため、道路の幅や建物の高さによって補間画像の精度が変化してしまうと考えられる。そのため、道路の幅や建物の高

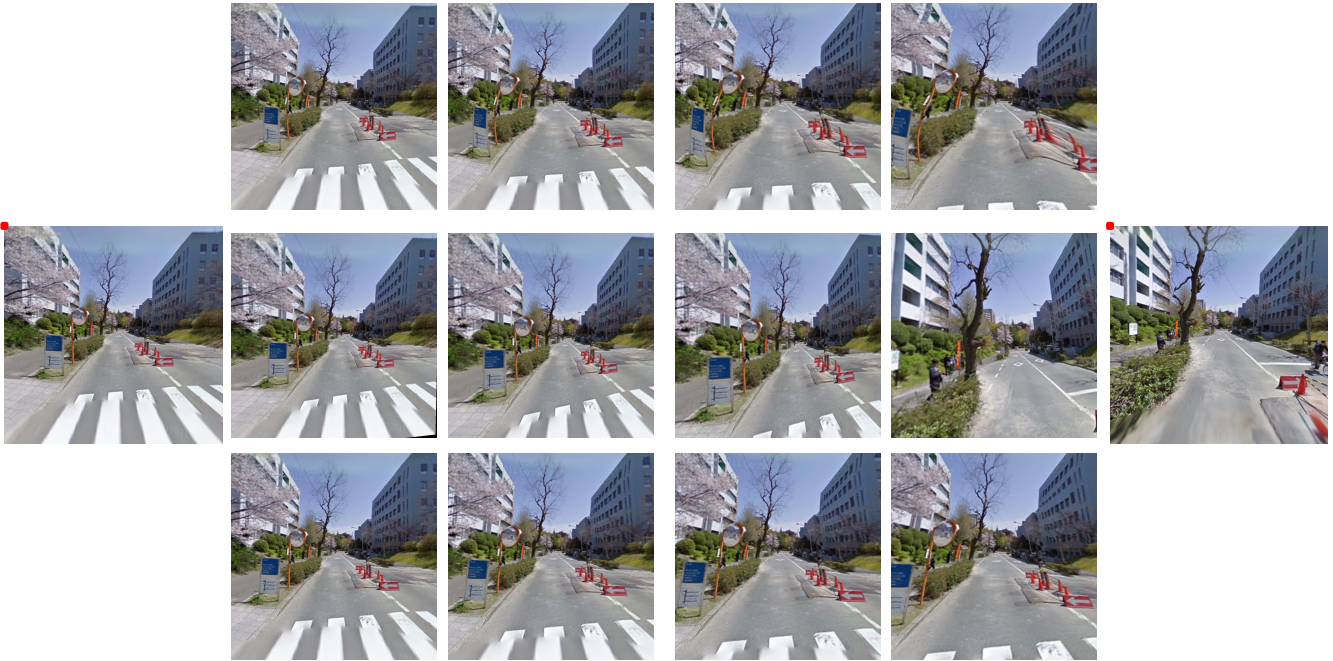


図 9: 赤枠の左の画像が画像 1, 右が画像 2 であるときに作成できる補間画像. 上は Thin-plate spline 法を使用し生成した補間画像, 中段は一つのホモグラフィ行列のみを使用し生成した補間画像, 下は提案手法により生成した補間画像である.

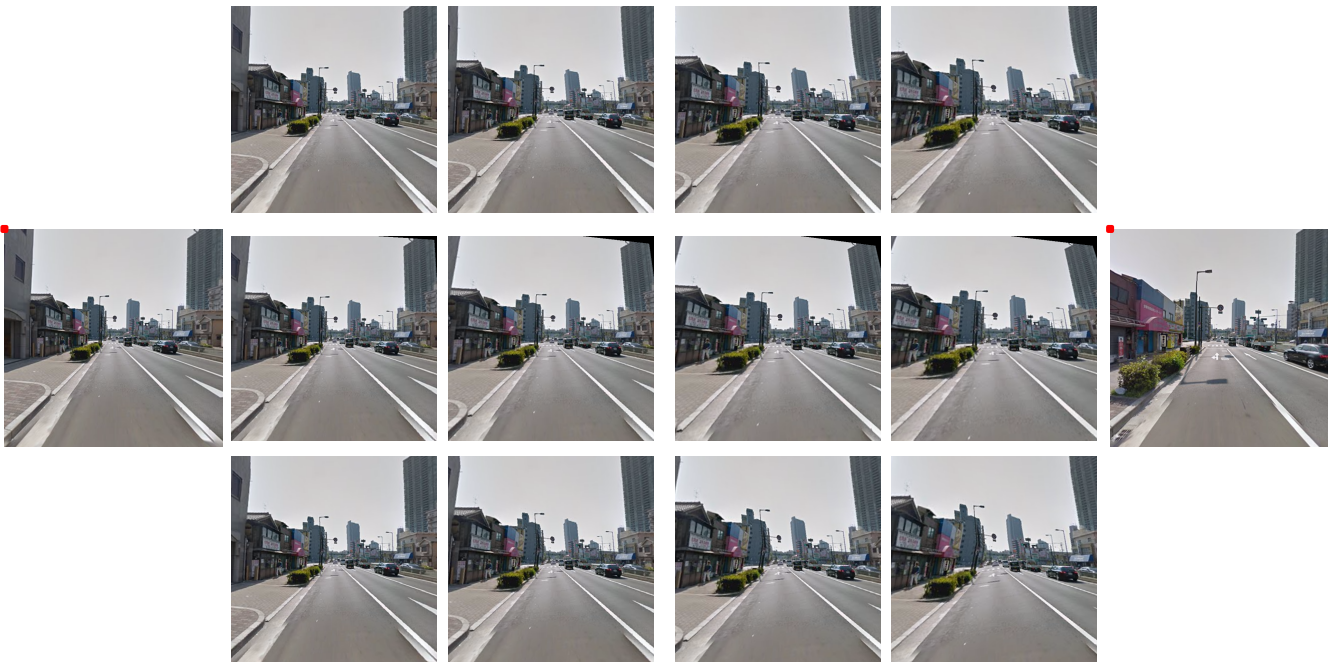


図 10: 赤枠の左の画像が画像 1, 右が画像 2 であるときに作成できる補間画像. 上は Thin-plate spline 法を使用し生成した補間画像, 中段は一つのホモグラフィ行列のみを使用し生成した補間画像, 下は提案手法により生成した補間画像である.

さを画像から推定し, それに基づいて境界線を引くことでさらに精度が向上すると思われる.

また, 道路の対応点探索について, 提案手法では SURF

のみを用いた対応点探索と比較し精度が上がっているが, 建物などに比べて対応点探索の精度が悪い. そのため, 道路においてはカメラとの位置関係が一定であるため対応点

を用いずに関連付けられている GPS シグナルから画像間の道路の変化を推測し、それに基づいて補間画像を作成するといった方法が有効であると考えられる。

参考文献

- [1] Kinsman, D.: Teehan+Lax Labs - Hyperlapse, <http://labs.teehanlax.com/project/hyperlapse>.
- [2] Gonzalez, R. C. and Woods, R. E.: *Digital Image Processing*, Prentice Hall, 3rd edition (2007).
- [3] Lucas, B. D. and Kanade, T.: An Iterative Image Registration Technique with an Application to Stereo Vision, *Proc. 7th International Conference on Artificial Intelligence* (1981).
- [4] Lowe, D. G.: Object recognition from local scaleinvariant features, *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference*.
- [5] Bay, H., Tuytelaars, T. and van Gool, L.: Surf: Speeded up robust features, *Computer VisionECCV 2006*, pp.404-417 (2006).
- [6] Choi, B.-D.: Motion-Compensated Frame Interpolation Using Bilateral Motion Estimation and Adaptive Overlapped Block Motion Compensation, *IEEE Circuits and Systems Society* (2007).
- [7] Seitz, S. M. and Dyer, C. R.: View Morphing, *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques pp.21-30* (1996).
- [8] Gurdan, T., Oswald, M. R., Gurdan, D. and Cremers, D.: Spatial and Temporal Interpolation of Multi-View Image Sequences, *Lecture Notes in Computer Science* (2014).
- [9] Harris, C. and Stephens, M.: A combined corner and edge detector, *Proceedings of Fourth Alvey Vision Conference*, pp. 147–151 (1988).
- [10] Seitz, S. and Dyer, C.: Physically-Valid View Synthesis by Image Interplation, *Proceedings of the IEEE Workshop on Representation of Visual Scenes, VSR '95*, Washington, DC, USA, IEEE Computer Society, pp. 18–(1995).
- [11] Yang, F., Shechtman, E., Wang, J., Bourdev, L. and Metaxas, D.: Face Morphing using 3D-Aware Appearance Optimization, *Proceedings of Graphics Interface 2012 pp.93-99* (2012).
- [12] Schempp, W. and Zeller, K.: *Constructive Theory of Functions of Several Variables*, Springer (1977).
- [13] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P. and Susstrunk, S.: SLIC Superpixels Compared to State-of-the-art Superpixel Methods, *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, num. 11, pp.2274 - 2282* (2012).