

# 実写画像を用いる仮想空間の生成

3S-2

亀井克之 丸山 稔

三菱電機（株）先端技術総合研究所

## 1. はじめに

環境入力の容易さやリアリティの高さから、仮想空間の生成に際して実写画像を利用する手法が研究されている。通常、限られた画像から、3次元形状の復元や画像間の領域対応付けを行って視点移動に対処する<sup>1) 2)</sup>が、それぞれに未だ困難が残る。本稿では、視線情報による画像の管理・合成によって視点移動に対処する手法を示す。画像の領域（あるいは画素）を視線で管理し、新たな画像は、視線が一致、もしくは近い既存画像の領域を組み合わせることで生成していく。

## 2. 視線による画像管理と合成方法

画像中の画素を指定すれば視線が定まる。新たな画像の生成は、各画素に対して視線を定め、これと同一の視線をもつ画素を既存画像から検索し、その値を画素値とすることによって行う（図1）。このとき、とりうる視線は3次元空間上の任意の直線となるが、そのすべてに画素を用意するのは不可能である。しかしながら、応用面を考えると、都市のウォークスルーなど2次元平面内の視点移動で十分な効果が上がる例も少なくない。そこで、以下ではこのような平面上での視点移動に限定して議論を進める。

## 3. 平面上での視点移動に対する画像生成

### 3.1 Hashed View による画像管理

視点移動を平面上に限定する場合について考える。このときも、とり得る視線は膨大となるが、上下方向については同一の既存画像を利用することによってこれに対処する（図2）。 $K$ 枚の実写画像  $I_k(i, j)$  ( $k=1, \dots, K, i=1, \dots, N, j=1, \dots, M$ 、また、視点  $p_k$  と視軸は  $xy$  平面上にあるとする) を幅1、高さ  $M$  の短冊型画像（以下 Hashed View と呼ぶ）に分解、おのおのを  $H_{ki}(j)$  とする。Hashed View に対する視線は、視点  $p_k=(x_k, y_k)$  から  $xy$  平面上 Hashed View に向かうベクトル（方向  $\alpha_{ki}$ ）で表す。このベクトルを、Hough 変換で用いられるパラメータ  $\rho, \theta$  によって、

$$\theta_{ki} = \alpha_{ki} - \pi/2$$

$$\rho_{ki} = x_k \cos \theta_{ki} + y_k \sin \theta_{ki} \quad (1)$$

と表現し、この集合  $V = \{(\rho_{ki}, \theta_{ki})\}$  により画像を管理する。

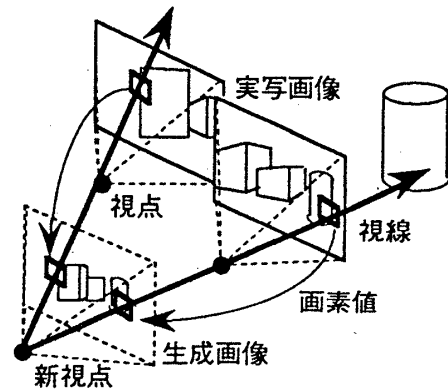


図1 実写画像を用いる画像合成方法

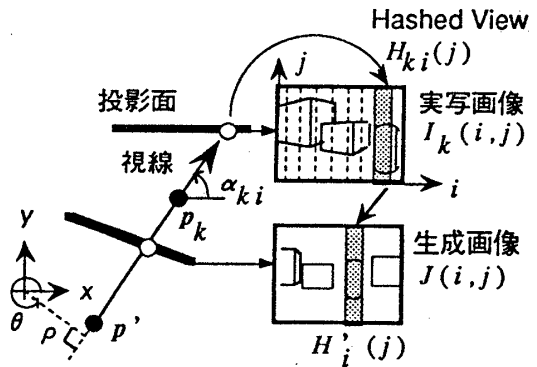


図2 Hashed View による画像管理と生成

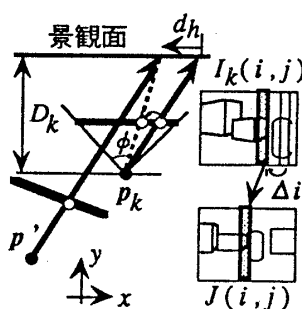


図3 水平方向の補正

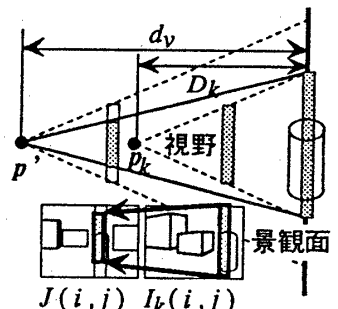


図4 上下方向の補正



図5 入力実写画像 (一部)



(a) (b)

図6 本手法による生成画像の例



図7 比較用実写画像

### 3. 2 画像生成と奥行き情報による補正

新たな視点  $p'$  が設定された場合、求める生成画像  $J(i, j)$  の各 Hashed View ( $H_{i'}(j)$ ) に対して (1) 式と同様に  $(\rho, \theta)$  を求め、これに合致する要素を  $V$  の中から検索する。検索された要素の  $k$  と  $i$  ( $k^*$ ,  $i^*$  とする) により合致する視線をもつ Hashed View を特定し、

$$H_{i'}(j) = H_{k^* i^*}(j) \quad (2)$$

として画像  $J(i, j)$  を生成していく。

このとき、検索された視線は完全に一致するとは限らず、また、上下方向は遠近法の効果を加味する必要があり、水平方向、上下方向それぞれに補正を行う。この補正には視点から対象までのおおまかな距離情報を利用する。 $k$  番目の画像について主たる対象 (都市景観の場合は例えば注目する建物) までの距離  $D_k$  に景観面を設定する。まず、水平方向については、図3に示すように景観面上での視線の差を  $d_h$  とし、

$$\Delta i = N d_h / \{2 D_k \tan(\phi/2)\} \quad (3)$$

として、視線上に適切な像が得られるように Hashed View を  $\Delta i$  だけ変更、これに合わせ  $i^*$  を更新する。ここで、 $\phi$  は視角である。上下方向は、遠近法の効果に合うように、 $p'$  から景観面までの距離  $d_v$  を求め、

$$H_{i'}(j) = H_{k^* i^*}(j^*) \quad (4)$$

$$j^* = (j - M/2) d_v / D_k + M/2$$

のように、 $j = M/2$  を中心に伸縮を施す (図4)。以上のようにして、歪みの低減を図る。

### 4. 画像生成結果

上で述べた手法を用いて、仮想空間を生成した。図5に使用した実写画像の一部を示す。ビデオカメラを用い、視軸に垂直に約20cmづつ平行移動して撮影した。図6はこれらを用いて合成した画像で、(a)は視点を後方に、(b)は前方に移動させた例である。それぞれ36枚、10枚の画像を組み合わせて合成されている。現実感の高い画像が生成できており、歪みも目立たない。生成・表示の処理時間は、平均0.05秒であった (画像サイズ320×240、計算機 IRIS Crimson RealityEngine)。図7に、比較のため図6(b)の視点位置で撮影した実写画像を示す。

### 5. おわりに

視線情報を用いて、実写画像から視点移動に対応した画像を生成する手法を示し、仮想空間を構成した。今後、自由な視点移動への対応、また、合成画像の画質向上について研究を進める予定である。

### 参考文献

- 1) 廣瀬, 佐藤: 人工現実感技術を用いた臨場感の合成とその伝達, 第9回ヒューマン・インタフェース・シンポジウム, 111-116 (1993).
- 2) S.E.Chen & L.Williams: View Interpolation for Image Synthesis, SIGGRAPH 93 Conference Proceedings, 279-288 (1993).