

リアリティの高い仮想空間構築のための 実写テクスチャの活用（2）

1H-7

—実行効率を考慮したテクスチャ画像量の調整—

森野崇志 大島登志一 山本裕之 田村秀行
キヤノン株式会社 情報メディア研究所

1. はじめに

我々の開発している仮想空間体験システム PreView は、仮想空間を作成するサブシステム (Creator) と仮想空間を擬似体験するサブシステム (Player) から成る。テクスチャマッピングを多用した仮想空間を Player で表示する際、現在のグラフィックワークステーションでは、一画面を描画するのに用いるテクスチャ画像量の総量がテクスチャメモリ容量を超過すると、描画効率が急激に低下するという問題がある。そこで、テクスチャのリアリティを落とさずに、仮想空間生成時にテクスチャ画像量を調整する機構 (Texture Tuner) を実現した。

2. Texture Tuner の機能と操作手順

Texture Tuner は、Creator で生成する仮想空間データに対し、マッピングすべきテクスチャ画像の解像度を変化させることにより、そのデータ量を調整するツールである。Texture Tuner の操作手順と機能の概略は以下の通りである。

(1) テクスチャ画像量の判定は、図 1 に示すような平面図に重畳した正方格子点毎に行う。各格子点で水平方向に離散的に視線方向を変化させて、各方向毎に描画に必要なテクスチャ画像のデータ量の和を計算する。格子の粗さ、視点位置の高さは可変であるが、このような離散的な位置と方向のみを取るのには、現実的な応答速度にするためである。

(2) 全格子点位置で全周を 4, 6, 8 方向 (選択可能) に視線を変化させ、いずれか一方向でもテクスチャメモリ容量を超過した位置が、図 1 のように図示される。

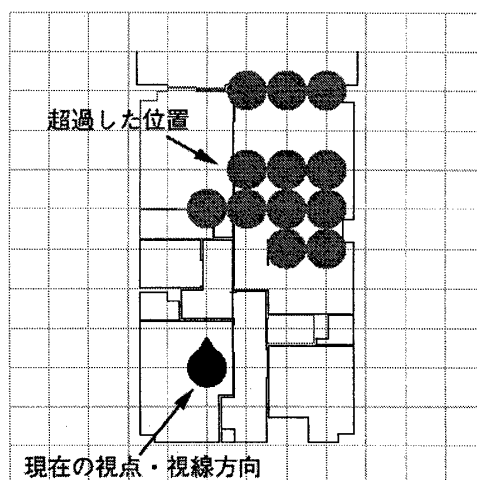


図 1 : 操作画面 GUI 例

(3) 超過量が最大となる格子点は、常に別の色で表示されるので、操作者はこれを手掛かりに後述の調整方法により、個別のテクスチャ画像の解像度 (i.e., 画像サイズ) を低下させ、総データ量を低減する。ここで、操作時の補助的な情報として、以下のものをグラフィック表示する機能を有している。

- ・テクスチャ画像をプレビューし、その画像サイズとデータ量を表示。
- ・現在の視点・視線方向で描画に用いるオブジェクトとテクスチャ画像のリストを表示。
- ・テクスチャ画像量の総量に対する各テクスチャ画像量の割合をチャート表示。

(4) テクスチャ解像度を落とした場合に、上述の格子点と視線方向でのレンダリング結果を表示する機能も備わっている。これによって、Player での本格的なウォークスルーに移行することなく、Creator 内で簡易離散空間移動して画質を確認できる。

(5) 次々に最大超過する格子点位置でこの操作を繰り返すことにより、全点でテクスチャメモリ容量以下にデータ量を抑制する。この離散的な位置・方向以外の任意の視点・視線は抑制が理論的に保証されていないが、実用上は問題ない。

Texture Mapping Technique for Creating Photo-realistic Virtual Space (2):—Texture Data Tuner Considering Rendering Performance—

Takashi MORINO, Toshikazu OHSHIMA, Hiroyuki YAMAMOTO, and Hideyuki TAMURA

Media Technology Laboratory, Canon Inc.,
890-12, Kashimada, Saiwai-ku, Kawasaki 211, Japan

3. 調整方法

テクスチャ画像量を調整するためには、各視点・視線方向のテクスチャ画像の総量を計算後、

- (1) どの視点・視線方向時のデータを調整するか
- (2) どのテクスチャ画像を調整するか
- (3) どの程度調整するか

を決定する必要がある。データ量削減候補の決定法としては、テクスチャ画像サイズ、もしくはマッピングされる平面（又は曲面）の面積の大きい順や、マッピング倍率の小さな順等が考えられる。しかし、マッピング結果の画質評価はテクスチャ画像の特徴（空間周波数等）に依存し、またユーザの好みにも左右される。

そこで本システムでは、以下の3つのモードを設け、ユーザがこれを選択できるようにした。

(a) 手動調整モード: 上記3項目を全てユーザが指定する。

Step1: 調整する視点・視線方向を選択する。

Step2: 調整するテクスチャ画像を、上記位置で使用されている画像から一つ選択する。

Step3: ユーザは選択テクスチャ画像量を原画像量に対する倍率で指定し、システムはこの倍率から調整量を決定する。

Step4: 再度各視点・視線方向でテクスチャ画像量計算後、Step1に戻り処理を繰り返す。

(b) 半自動調整モード: テクスチャの調整量はシステムが自動的に決定する。

Step1: 調整する視点・視線方向を選択する。

Step2: 上記位置で使用されているテクスチャ画像から、調整する画像を $I_i (i=1, \dots, M)$ 、調整の対象としない画像を $I_i (i=M+1, \dots, N)$ とする。

Step3: 次式を満たす最大の $\alpha (0 < \alpha < 1)$ を求める。

$$M \geq \alpha \sum_{i=1}^M D_i + \sum_{i=M+1}^N D_i \quad (1)$$

ここで L はテクスチャメモリ容量、 D_i は画像 I_i のデータ量である。

Step4: 求めた α を用いて、画像 $I_i (i=1, \dots, M)$ をそれぞれ α 倍にスケールアップする。

Step5: 再度各視点・視線方向でテクスチャ画像量計算後、Step1に戻り処理を繰り返す。

(c) 自動調整モード: 上記3項目を全てシステムが指定する。

Step1: 調整する視点・視線方向を、テクスチャ画像量の総量の超過量が最大である位置とする。

Step2: 調整するテクスチャ画像を、上記位置で使用されている全画像 $I_i (i=1, \dots, N)$ とする。

Step3: 式2を満たす最大の $\alpha (0 < \alpha < 1)$ を求める。

$$M \geq \alpha \sum_{i=1}^N D_i \quad (2)$$

Step4: 求めた α を用いて、画像 $I_i (i=1, \dots, N)$ をそれぞれ α 倍にスケールアップする。

Step5: 再度各視点・視線方向でテクスチャ画像量計算後、超過位置が無くなるまで繰り返す。

この自動調整後にさらにユーザが手動で調整を加えることも可能とした。

4. 実験結果

文献[1]に示した仮想建築物に対して、調整前の仮想空間データ（テクスチャメモリを超過したデータ）と本 Texture Tuner によりテクスチャメモリ以下に調整したデータとの描画性能比較実験を行った。

● 使用環境

機種: SGI IRIS Crimson / RealityEngine

テクスチャメモリ容量: 4Mbytes

● 実験結果

この実験では、調整後のデータにより提示された仮想空間に、リアリティの劣化はほとんど認められなかった。描画スピードに関しては、表1に示すように大幅な実行効率向上が達成されている。

表1: 実験結果

使用データ	フレームレート
調整前(6~7MB)	0.8~1.2Hz
調整中(4~4.5MB)	1.5~2.0Hz
調整後(3~4MB)	7.0~9.0Hz

5. まとめ

リアリティの高い仮想空間を提示する際には、ハードウェア等の制限を考慮した実行効率向上のための処理が必要である。本 Texture Tuner では、使用テクスチャ画像量の調整を対話的に行うことにより実行効率の向上を図っている。今後は、テクスチャ画像の空間周波数や明度等を活用した、より効率的な調整アルゴリズムの検討を進める予定である。

参考文献

- [1] 山本, 内山, 大島, 田村: リアリティの高い仮想空間構築のための実写テクスチャの活用(1) - 仮想空間構築ツール -, 第52回情報処全大, 1H-06(1996).