

法政大学学術機関リポジトリ

HOSEI UNIVERSITY REPOSITORY

PDF issue: 2025-05-10

実物体の表面光学特性を考慮したプロジェクト タによる内部透過映像の重畳表示

UEKUSA, Naoto / 植草, 直人

(出版者 / Publisher)

法政大学大学院情報科学研究科

(雑誌名 / Journal or Publication Title)

法政大学大学院紀要. 情報科学研究科編 / 法政大学大学院紀要. 情報科学研究科編

(巻 / Volume)

10

(開始ページ / Start Page)

1

(終了ページ / End Page)

6

(発行年 / Year)

2015-03-24

(URL)

<https://doi.org/10.15002/00011698>

実物体の表面光学特性を考慮したプロジェクタによる内部透過映像の重畳表示

Superimposed Projection of Ghosted View on Real Object with Color Correction

植草 直人*
Naoto Uekusa

法政大学大学院 情報科学研究科 情報科学専攻
e-mail: naoto.uekusa.7m@stu.hosei.ac.jp

Abstract

We describe a spatial augmented reality system that enables superimposed projection of an internal image on a real object with color correction. Our system is a projector-camera system, which consists of a camera, a projector, and a PC. At first, we generate a first projection image from the internal image of CG and a camera image of the real object captured by the camera. The first projection image is generated to avoid the pattern of the real object by importance map which is generated camera image. Next, we project the first projection image on the real object, and again capture an image of the real object with the internal image. At last, we update the projection image with color correction on CIELUV color space and project the image on the real object. This system will be able to visualize the internal structures on various objects easily because this system is reproduced on real object enhances a sense of depth. By using this effect, our system are enable to use various application such as support of work by the visibility of the complex internal structure.

1 はじめに

本研究では、我々は、内部透過画像を実物体に重畳投影表示する空間型拡張現実感 (Spatial Augmented Reality) [3] システムを提案する。内部透過画像とは、X線CT画像のような一般的な内部構造の画像に加え、実物体の内部が透けて見えているような効果を付加した画像である。重畳投影とは、プロジェクタを用いて、外壁面や立体物などの、スクリーンのような理想的な反射特性を持たない物体に対して、投影するときに用いられる技術である。我々は、この内部透過画像と重畳投影を用いて、内部透過画像を実物体に重畳投影することで、物体の透過を再現し、ユーザーがデバイスの装着せずにAR体験できるシステムを開発した。内部投影画像は、投影対象物の色に応じた色補正を行うことで、より正確な表現を目指す。

内部透過表現に関しては、Zollmannら[12]が2次元画像に対する“Ghosted View”を提案している。この手法は、カメラ画像から、透過率を求めるための重要度マップを計算し、その重要度マップから、計算した透過率

とCGモデルの画像を合成して内部構造の透過画像を生成し、実物体画像の透過表現を再現している。Kalkofenら[8]は、3Dモデルの曲率、実物体画像とその重要度マップ、内部3Dモデル形状を用いて、立体物に対しての“Ghosted View”を提案している。重畳投影に関しては、Raskerら[10]がCGのプロジェクションマッピングとして、“Shader Lamps”を提案している。カメラを用いた投影画像の制御に関する研究として、Nayarら[9]が、カメラの画像を事前に決定した情報に一致させる光学補償を再現している。

上記“Ghosted View”は、画像生成技術のみの提案のため、タブレット端末等のディスプレイデバイスの使用が必須である。また実物体上での投影は多様な場所での利用を考慮する必要があるため、照明や実物体の色の影響に強い投影画像の補正を行う必要がある。提案手法を用いることで、利用者が、建築物の内部配管や配線、機械の内部構造を容易に確認ができるため、修復作業の補助や学習など応用先があり、内部構造の容易な変更や、動作のある内部構造の表現も可能になるという利点がある。

本稿では、2章で提案手法のフローチャート、3章で目標画像と初期投影画像の生成、4章で投影画像の補正処理について述べる。5章で提案手法の実験結果について述べ、最後にまとめを述べる。

2 提案手法

実物体の内部透過を実現するために、我々は箱の中にあるものを観測者がデバイスを使わずに箱が透けて中身が見えるという状態を再現する手法を提案する。提案する手法として、投影対象の箱の模様とCGモデルが重ならないようにする、投影対象の箱の色に応じた投影画像の色補正を行うという2点を実装する。この2点を実装することで観測者に内部透過画像の奥行き感を観測者が理解できるようにすることができ、箱の中に中身があるように感じさせることができる。提案手法のフローチャートを図1に示す。初めに、Kinectで取得した投影対象物である箱を撮影した画像(図1のCamera Image)と、プロジェクタ視点からレンダリングした箱の内部と箱の内壁が描かれたCGモデルの画像(図1のCG Image)を使い、画像補正のための目標画像(図1のGoal Image)と初期投影画像(図1のFirst Projection Image)を生成する。続けて、箱に合わせて投影画像のア

*Supervisor:Takafumi Koike, Professor

フィン変換を行い、初期投影画像を投影する。そして初期投影画像の投影後の箱の画像と目標画像の比較から、初期投影画像の調整を行い、調整した画像(図1の Final Projection Image)を改めて箱に投影することで実物体上での内部透過の再現をおこなう。図1の目標画像と初期画像の生成 (Generate Goal Image and First Projection Image) と投影画像の補正処理 (Generate Final Projection Image) については次節で述べる。

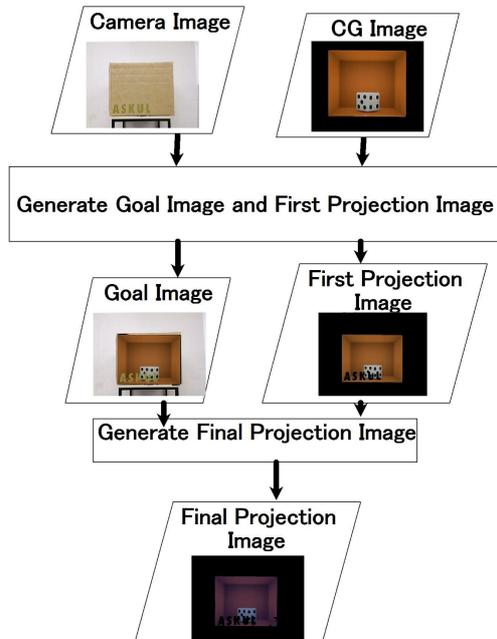


図1: 提案手法のフローチャート

2.1 目標画像と初期投影画像の生成

目標の内部透過画像と初期投影画像の生成処理の流れを図2の左の領域に示す。2つの画像の生成にはCGモデル, Kinectで取得した投影対象物を撮影した画像を用いて目標画像の生成と初期投影画像の生成を行う。この処理は投影対象物に描かれている文字, 模様, 輪郭部分にCGが重なることを避け, 模様の奥側にCG画像があるようにプロジェクタで投影する。撮影した画像から透過率を計算するための重要度マップ(図2の Importance Map)を使うことで, 実物体の模様部分を除いた部分の透過率を計算し, それをCGモデル画像に反映することで求める初期投影画像と, カメラ画像とCGモデルの合成によって求める内部透過の目標画像を生成する。この処理の詳細は3章で述べる。

2.2 投影画像補正処理

投影画像補正処理の流れを図2の右の領域に示す。最初に初期投影画像を投影した投影対象物が撮影された画像(図2の Camera Image after Projection)と目標画像をCIELUV色空間[11]の値で比較し, (L, u, v) の値が異なる点の x, y 座標に対応した初期投影画像上の点の位置を求める。そして投影前の投影対象物の画像(図2の Camera Image)のCIELUV色空間の値を基に, 初期投影画像の (L, u, v) の値を再計算する。そして補正後の投影画像を投影対象物に投影することで, 内部透過画像を実

物体上に再現をする。この処理によって物体の色や照明の影響に強いCG画像の投影を行う。この処理の詳細は4章で述べる。

3 目標画像と初期投影画像の生成

Zollmanらの手法[12]とKalkofenらの手法[8]に基づいて, 目標画像と初期投影画像を生成する。2つの画像の生成には, カメラ画像の画素ごとの透過率計算が必要となる。画素ごとの透過率計算には, 重要度マップ $M(x, y)$ を用いて計算する。本研究では, 初めに, 重要度マップを生成し, 続いて, $M(x, y)$ から物体の透過率 $\alpha(x, y)$ を計算する。そしてKinectからリアルタイムで得たカメラ画像 $O(x, y)$ とCG画像 $C(x, y)$ と物体透過率 $\alpha(x, y)$ から内部透過画像 $G(x, y)$ を生成する。図2の重要度マップの生成処理 (Generate Importance Map) と目標画像と初期投影画像の生成 (図2の Generate Goal Image and First Projection Image) の処理について次節で述べる。

3.1 重要度マップの生成

画素ごとの透過率の計算のために重要度マップを求める。重要度マップの求め方のフローチャートを図3に示す。

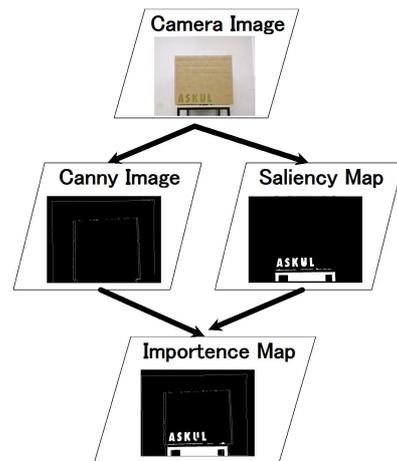


図3: 重要度マップ $M(x, y)$ 生成処理のブロック図

カメラ画像 $O(x, y)$ から重要度マップ $M(x, y)$ を生成するため, カメラ画像にCanny edge[4]フィルタをかけた画像 $E(x, y)$, カメラ画像からSaliency Image[7]を求めて, 求めた画像に対して領域分割を行い, その領域ごとに二値化を行った画像 (Saliency Map) $S(x, y)$ を中間画像として生成する。

各中間画像を用いて, 式1の条件で重要度マップ $M(x, y)$ を求める。

$$M(x, y) = \begin{cases} E(x, y), & \text{if } E(x, y) > 0 \\ S(x, y), & \text{elseif } S(x, y) > 0 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

重要度マップの白い部分 $M(x, y) = 1.0$ の部分が投影画像と重ならない領域となり, 重要度マップの黒い部分 $M(x, y) = 0.0$ の部分が透過処理を用いた投影画像と箱が重なる領域となる。本論文ではSaliency Map $S(x, y)$ の生成手法について述べる。

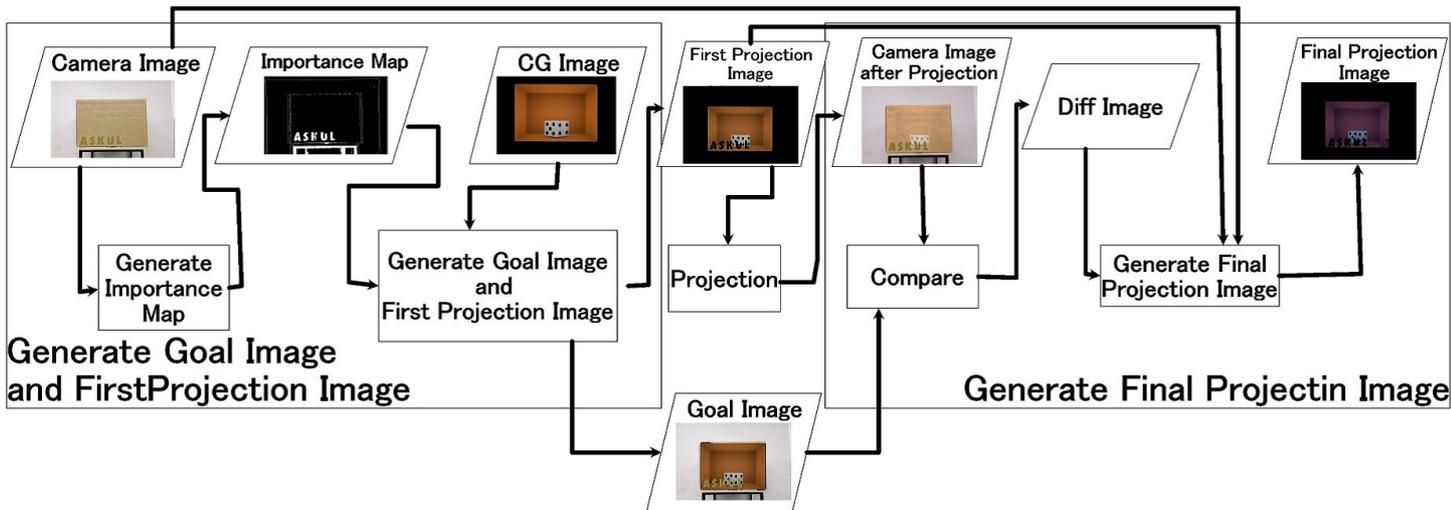


図 2: 目標画像と初期投影画像の生成, 投影画像補正処理の流れ

3.1.1 SaliencyMap $S(x, y)$ の生成

ここで, Achanta らの手法 [1] をもとに $S(x, y)$ の生成方法を図 4 示す.

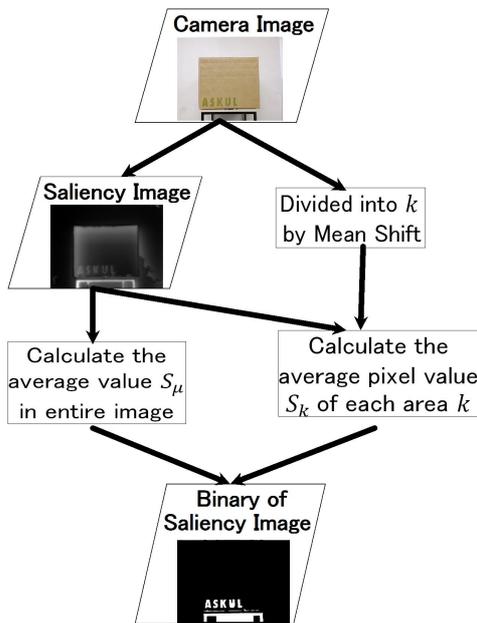


図 4: $S(x, y)$ の生成処理図

初めに, カメラ画像 $O(x, y)$ の輝度値, 色情報から Saliency Image(図 4 の Saliency Image) を生成する. 続いて, カメラ画像 $O(x, y)$ を Mean-shift 法 [5] を用いて領域分割する. $O(x, y)$ を Felzenszwalb ら [6] の手法を基に k 個にクラスタリング, 各領域の画素の平均値 S_k を求める. 最後に, Saliency Image 全体の画素平均値 S_μ を基準に S_k を二値化したものを $S(x, y)$ (図 4 の Binary of Saliency Image) とする. $S(x, y)$ は物体内の文字や模様部分を抽出できているため, その部分に投影画像が重ならないようにすることができる.

3.2 目標画像と初期投影画像の生成

内部透過画像 $G(x, y)$ 生成のため, 物体透過率 $\alpha(x, y)$ を決める必要がある. 物体透過率は重要度マップ $M(x, y)$ と初期透過率 t から次式を用いて物体透過率 $\alpha(x, y)$ を求める.

$$\alpha(x, y) = t \times (1.0 - M(x, y)). \quad (2)$$

次に, 物体透過率 $\alpha(x, y)$, カメラ画像 $O(x, y)$ と CG 画像 $C(x, y)$ を用いて式 3 に示すように合成し, 初期内部透過画像 $G(x, y)$ を生成する.

$$G(x, y) = (1.0 - \alpha(x, y)) \times O_0(x, y) + \alpha(x, y) \times C(x, y) \quad (3)$$

$G(x, y)$ の画像の結果から, CG 部分 ($\alpha(x, y) \times C(x, y)$) を取り出した投影画像 $P_0(x, y)$ を生成する. 処理結果後の投影画像 $P_0(x, y)$ を図 5 に示す. そして画像 $P_0(x, y)$ にアフィン変換をかけて, 投影対象の実物体に合わせて投影する.

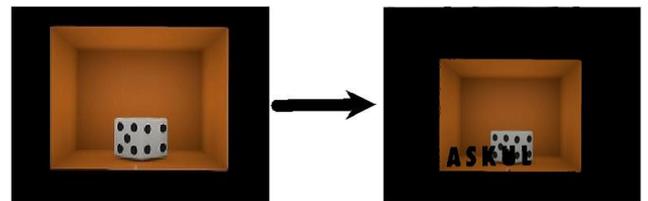


図 5: CG 部分を取り出した投影画像 P_0

4 投影画像補正処理

実物体への投影は通常のスクリーン上への投影と異なり, 物体本来の色, 文字や模様, 物体固有の反射率, 白色照明下の影響を受けるため, 見かけに応じて投影画像の補正を行う必要がある. 本稿では, 透過映像を投影後に, カメラから得た情報を用いて, 実物体本来の見かけに応じた投影画像の調整, 制御を行う手法を用いる.

画像補正の手法は Ashdown らの手法 [2] を基づいて、投影画像の補正を行う。

4.1 色空間の変更と画像の比較

投影後の実物体画像と目標画像の比較のために、2つの画像を CIELUV 色空間に変換する。CIELUV 色空間への変換は、RGB 画像を CIEXYZ 色空間 [11] の画像に変換した後、さらにその XYZ 色空間の画像から式 4 を用いて CIELUV 空間に変換する。なお L, u, v の値の範囲はそれぞれ $0 \sim 1.0$ とする。

$$\begin{aligned} L &= 1.16(Y/Y_n)^{1/3} - 0.16 \\ u &= 13L(u' - u'_n), \quad u' = 4X/(X + 15Y + 3Z) \\ v &= 13L(v' - v'_n), \quad v' = 9Y/(X + 15Y + 3Z) \end{aligned} \quad (4)$$

上記の式で色空間を変更した後、カメラ画像と目標画像の比較を行う。処理の高速化のため 3 章で求めた重要度マップによって生成した初期投影画像を用いて文字、模様の部分 ($M(x, y) = 1.0$) を除いた位置の画素を比較対象とする。(L, u, v) の値が異なる点の x, y 座標に対応した $P_0(x, y)$ 上の位置 (x_1, y_1) と (L, u, v) の値が異なる点の個数を $pixels$ として計算する。画像 $P_0(x, y)$ 上の (x_1, y_1) の位置にあたる画素の値を $P_0(x_1, y_1) = (L_0, u_0, v_0)$ とし、 $P_0(x_1, y_1)$ の点を色彩の調整を行う。

4.2 色彩の調整

投影画像の色彩の調整手法として、4.1 で求めた $P_0(x_1, y_1)$ の uv 値 (u_0, v_0) を調整する式 5 を使い、式 5 内の (s, a, b) を変えていくことで色彩の調整を行う。

$$\begin{bmatrix} u_1 \\ v_1 \end{bmatrix} = s \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} + \begin{bmatrix} a \\ b \end{bmatrix} \quad (5)$$

(u_1, v_1) を決める値 (s, a, b) を求める手法として、図 6 のような CIELUV 色空間を 2 次元上の uv 空間として考え、RGB 空間上の赤、マゼンダ、青、シアン、緑、黄色にあたる点から図 6 のように、六角形を形成する 6 本の直線引き、図 1 の Camera Image を CIELUV 空間に変化した値と 6 本の直線の距離関係から (s, a, b) を求める。

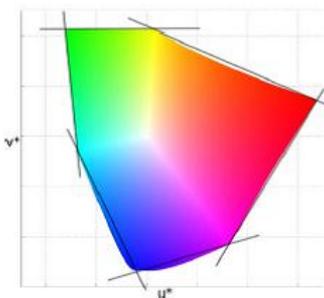


図 6: uv 空間図 ([2] より引用)

初めに、上記の 6 本の直線を $l_a u_0 + l_b v_0 + l_c = 0$ として定め、 $(l_a, l_b, l_c)^T$ (正し、 $l_a^2 + l_b^2 = 1.0$ とする) を求める。そして直線と $P_0(x_1, y_1)$ との点と直線の距離の関係

と距離の式と式 eq:6 から、 $l_a(u_0 s + a) + l_b(v_0 s + b) + l_c$ と表す。さらにベクトルを用いた表現、 $\mathbf{r} \cdot \mathbf{m} + l_c$ ($\mathbf{r} = (l_a u_0 + l_b v_0, l_a, l_b)$, $\mathbf{m} = (s, a, b)^T$) の形に変える。そして時間 k の時の \mathbf{m} を用いて、 $k + 1$ のときの \mathbf{m} の値を式 6, 7 で求める。

$$\mathbf{m}_{k+1} = \mathbf{m}_k - \gamma [HE(\mathbf{m}_k)]^{-1} \nabla E(\mathbf{m}_k) \quad (6)$$

$$E = c_1(1 - s)^2 + c_2(a^2 + b^2) + \frac{1}{n} \sum_{pixels} \sum_{lines} e^{c_3(\mathbf{r} \cdot \mathbf{m} + l_c)} \quad (7)$$

E はカメラ画像 O から求めた通常のスクリーン上の投影と、投影後の実物体画像との誤差を表した関数であり、 $HE(\mathbf{m}_k)$ は E のヘッセ行列、 $\nabla E(\mathbf{m}_k)$ は E の勾配。初期値として $\gamma = 1$, $lines = 6$, $n = pixels \times lines$, $c_1 = 3$, $c_2 = 20$, $c_3 = 4$, \mathbf{m}_0 の時の (s, a, b) の初期値を $s = 1.0$, $a = 0.0$, $b = 0.0$ とし、式 6 で求めた \mathbf{m}_{k+1} を用いて 5 の式から (u_1, v_1) を求める。

4.3 輝度範囲の決定

4.2 で求めた値 (u_1, v_1) に応じた L_1 の値を計算するために、 (u_1, v_1) の時の表示輝度の最大値と最小値 G_h, G_l を求める。 G_h, G_l は、4.2 で用いた RGB 空間上の赤、マゼンダ、青、シアン、緑、黄色にあたる点から形成した六角形 6 点と RGB 空間上の白点と黒点 (図 7 の六角形の中心の点) からなる、図 7 のような uv 空間図上に三角形を作り、図上で (u_1, v_1) の点が、どの三角形内にあるかで場合分けを行い、 G_h, G_l を決定する。

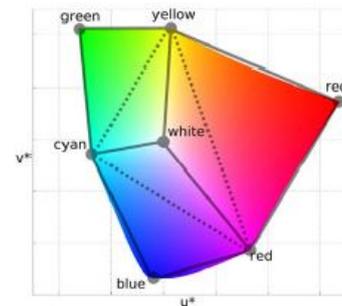


図 7: 図 6 を細分化したものの ([2] より引用)

図 7 の三角形の内側に (u_1, v_1) がある場合、その三角形を形成する 3 点を (L_{c1}, u_{c1}, v_{c1}) , (L_{c2}, u_{c2}, v_{c2}) , (L_{c3}, u_{c3}, v_{c3}) とする。ただし白点と黒点が三角形を形成する 3 点に含まれる場合、 G_h を求める場合は $L_{cn} = 1.0$, G_l を求める場合は $L_{cn} = 0.0$ とする。 G_h, G_l は 3 点の L の値の平均値から求める。図 7 の図形の外側に (u_1, v_1) がある場合は、図 7 を形成する 6 点から (u_1, v_1) の点に一番近い点の L の値を G_h, G_l とする。

4.4 輝度の調整

4.3 で求めた G_h, G_l から、 P_{x_1, y_1} 上の画素の輝度値 L_0 を (u_1, v_1) に応じた輝度値 L_1 の値を式 8 で計算する。

$$L_1 = F^l + (F^h - F^l)L_0 \quad (8)$$

F^l, F^h は、元画像の点 P_{x_1, y_1} の輝度値の表示限界で、初期値を $F^l = 0.0$, $F^h = 1.0$ とし、4.3 の処理で求めた G_h, G_l の値から Ashdown らの手法を基に F_l, F_h を変化させていき、式 9, 10 を用いて F^l, F^h を求めて、調整後の輝度値 L_1 を求める。

$$F^l = \overline{F^l} - \frac{d_1}{2} \frac{\partial r}{\partial F^l} - \frac{d_2}{2} \frac{\partial t}{\partial F^l} - \frac{d_3}{2} \frac{\partial w}{\partial F^l} \quad (9)$$

$$F^h = \overline{F^h} - \frac{d_1}{2} \frac{\partial r}{\partial F^h} - \frac{d_2}{2} \frac{\partial t}{\partial F^h} - \frac{d_3}{2} \frac{\partial w}{\partial F^h} \quad (10)$$

$\overline{F_l}, \overline{F_h}$ は、 P_{x_1, y_1} を中心に 8 のようなフィルタを用いて、 P_{x_1, y_1} の周囲の値の F_l, F_h の値を $\overline{F_l}, \overline{F_h}$ を求める。

| | | | |
|----------------|---|---|---|
| | 1 | 4 | 1 |
| $\frac{1}{20}$ | 4 | 0 | 4 |
| | 1 | 4 | 1 |

図 8: $\overline{F_l}, \overline{F_h}$ のフィルタ

式 9 と 10 で用いた値 r, t, w は、式 11, 12, 13 で求める。

$$r(L_1) = \begin{cases} (L_1 - G^l)^2, & \text{if } L_1 < G^l \\ (L_1 - G^h)^2, & \text{else if } L_1 > G^h \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

$$t = (F^l)^2 + (F^h - 1)^2 \quad (12)$$

$$w = e^{(F^l - F^h)} \quad (13)$$

各定数として $d_1 = 0.01, d_2 = 0.0001, d_3 = 0.0001, d_4 = 6.0$ とする。

4.5 補正画像の投影

4.2, 4.3, 4.4 で計算した投影画像 (L_1, u_1, v_1) を CIELUV 色空間から RGB 色空間に変換する。そして変更した画像を実物体に対して投影する。画像補正による投影画像の見かけの変化を図 9 示す。

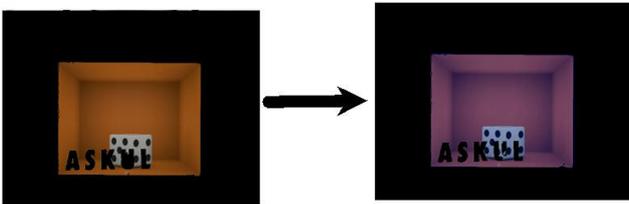


図 9: uv 補正を行った投影画像 P_n

投影後も 4 章の処理を行い (s, a, b) , (F_h, F_l) の値を計算と、カメラ画像と目標画像の比較を行い、物体上での内部透過画像の再現を行う。

5 実験

5.1 システム構成と実験環境

実物体上の内部透過画像再現のため、図 10 の構成を持つプロジェクタ-カメラシステムを構築した。

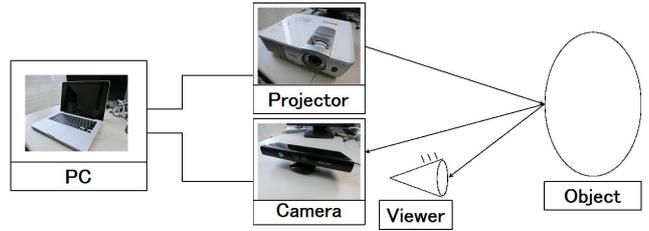


図 10: システム構成図

図 10 のようなシステム構成を用いて、11 左のような柄つきの段ボール箱を投影対象物 (図 10 の Object) とし、箱の文字が書かれている面に投影画像があたるように箱の位置を固定、提案手法に基づき実験を行った。投影対象物とカメラとの距離は 70cm、カメラはカメラの中心に投影対象物が収まるような高さに設置した。プロジェクタはカメラの後ろに設置し、投影画像が投影対象物にあたるように位置を合わせた。PC は MacBook Pro 13 インチ (CPU: Intel Core i5, メモリ: 4GB)、カメラは Kinect for Xbox 360、プロジェクタは BenQ の MW 817ST を使用した。初期透過率 $t = 0.8$ に定め、CG モデルは図 11 右のようなサイコロと箱の内壁が描かれた CG 画像を使用した。以上の条件で重要度マップを用いた投影画像による実物体の見え方、CIELUV 空間上での補正後の投影画像による実物体の見え方の 2 つの画像から評価した。



図 11: 左) 投影対象物, 右) 投影画像

5.2 重要度マップを用いた投影画像による実物体の見え方

重要度マップにより生成した投影画像による実物体の見え方の評価のため、通常の透過処理のみをおこなった画像と、図 5 で示した初期投影画像の 2 種類を実物体にそれぞれ投影し、見かけの比較を行った。プロジェクタと同じ視点から撮影した実物体の画像を図 12 に示す。文字部分のサイコロの目の色が投影されなくなり、箱の色と混ざった色領域がなくなることができた。この処理を行ったことで、サイコロの手前側に文字があるように表現された。



図 12: 左) 初期 CG 画像, 右) 3 章で生成した投影画像

5.3 CIELUV 空間上での補正後の投影画像による実物体の見え方

CIELUV 空間上での補正後の投影画像による実物体の見え方の評価のため、図 5 で示した透過画像の投影と図 9 で示した補正後の画像の 2 種類を実物体にそれぞれ投影し見かけの比較を行った。プロジェクタと同じ視点から撮影した実物体の画像を図 13 に示す。画像補正前と比べ、投影画像が濃い色になり、CG で描かれた箱の内側の壁が補正前の投影と比べて強調されるようになった。



図 13: 左) 3 章で生成した画像, 右) 4 章で生成した画像

6 考察

5.2, 5.3 の結果により、通常投影と比べ、投影画像の奥行き感が向上、箱の内側の壁が補正前の投影と比べて強調されるようになったことが確認できた。奥行き感の向上は重要度マップによる文字部分に CG が重ならないことで、サイコロの手前に文字が写っているように見え、サイコロと箱の表面の位置関係がわかりやすくなったことによると考えられる。内壁が強調されるようになったことは、実物体上への投影のための色補正を行ったことにより投影画像の色と実物体上の色が混ざることによって目標画像に近い画像になったためと考えられる。しかし、色補正後の画像のサイコロに元画像と比べ、青みがかかった色になってしまったため、内壁とサイコロを別々に画像補正をするなどの処理が必要と考える。

7 結論

実物体に内部透過 CG 画像を重畳表示することによって、実物体の内部透過表現を可能とする空間型 AR システムを提案した。実験の結果、実物体の表面、箱の内容物、箱の奥壁の位置関係がはっきりとわかるようになり、実物体の光学特性を考慮した内部透過映像表現できるようになったため、通常の投影と比べてより正確な表現を持った内部透過表現を実物体上へ重畳投影するシステムが実現できた。

しかし、現システムでは重要度マップの生成のために、実物体の色や光の影響を大きく受けるため、安定した計算が難しいことが課題である。そのため各種光の影響を受けない重要度マップの生成が必要である。また投影画像の補正手法については投影対象物が立体であるため、投影面が平面でない場合を考慮した場合の投影画像と目標画像の比較や内容物と箱の内壁の 2 つに分けて画像補正するなど他の画像補正の手法などについて考慮する必要がある。

今後は、上記の課題を解決するとともに、処理時間の長さ、実物体の形状が変化するような物体への対応、異なる視点での投影のための複数台プロジェクタ投影への対応などを行い、より実用性の高いシステム構築を目指していきたいと考えている。

参考文献

- [1] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pages 1597–1604, 2009.
- [2] M. Ashdown, T. Okabe, I. Sato, and Y. Sato. Robust content-dependent photometric projector compensation. *CVPRW '06*, 2006.
- [3] O. Bimber and R. Raskar. Spatial augmented reality merging real and virtual worlds. 2005.
- [4] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, pages 679–714, 1986.
- [5] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *PRMI*, 2002.
- [6] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2), pages 167–181, 2004.
- [7] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid analysis. *PAMI*, pages 1254–1259, 1998.
- [8] D. Kalkofen, E. Veas, S. Zollmann, M. Steinberger, and D. Schnalstieg. Adaptive ghosted views for augmented reality. *ISMAR 2013 Science and Technology Proceeding*, pages 1–4, 2013.
- [9] S. K. Nayer, H. Peri, M. D. Grossberg, and P. N. Belhumeur. A projection system with radiometric compensation for screen imperfections. *PROCAMS 2003*, 2003.
- [10] R. Raskar, G. Welch, K.-L. Low, and D. Bandyopadhyay. Shader lamps: Animationreal objects with image-based illumination. *2th EGSR*, pages 25–27, 2001.
- [11] G. Wyszecki and W. S. Stiles. Color science: Concepts and methods, quantitative data and formulae. (2nd ed. Wiley), 1982.
- [12] S. Zollmann, D. Kalkofen, E. Mendez, and G. Reitmayr. Image-based ghosting for single layer occlusions in augmented reality. *ISMAR 2010 Science and Technology Proceeding*, pages 13–16, 2010.