

科学研究費助成事業 研究成果報告書

平成 27 年 6 月 18 日現在

機関番号：22604

研究種目：基盤研究(C)

研究期間：2011～2014

課題番号：23611031

研究課題名(和文) デザイン検討用3次元CGソフトの開発と産業界への普及展開

研究課題名(英文) Development of 3D Computer Graphics software for design study and spread to design industry

研究代表者

笠原 信一 (Kasahara, Shinichi)

首都大学東京・システムデザイン学部・教授

研究者番号：00433178

交付決定額(研究期間全体)：(直接経費) 2,800,000円

研究成果の概要(和文)：デザイン分野における3次元CG活用の裾野を広げ、業界全体のデザインの質の向上や競争力の向上を図るために、デザイン検討をターゲットとし高品質と使いやすさを両立するCGレンダリング技術を研究開発し、独自ソフトウェア「Frend」を実現させた。さらにインターネットを介してそのソフトウェアを普及展開する仕組みを構築した。

3次元CGレンダリング技術の研究開発においては、間接光シミュレーション機能などのアルゴリズムを研究開発し実装することによって高品質表現を追求すると共に、膨大する計算時間を解消するために点光源による半影表現機能やGPGPU化などの処理高速化手法を研究し使いやすさの向上も目指した。

研究成果の概要(英文)：In order to use 3 dimensional Computer Graphics more widely and effectively in the design field and aim to improve design quality and competitiveness, CG rendering technologies for providing both the highest quality and easy of use was researched and finally the original software called "Frend" has developed. Further a system to spread the software through internet was developed.

At the stage to research CG rendering technologies, the highest quality was pursued by researching algorithm on indirect lighting simulation and so on. On the other hand, calculation acceleration algorithm, namely hemi shadow by a point light, GPGPU and so on, was also researched to solve the enormous calculation time to pursue ease of use.

研究分野：コンピュータグラフィックス

キーワード：コンピュータグラフィックス ソフトウェア デザイン

1. 研究開始当初の背景

デザイン分野における3次元コンピュータグラフィックス(CG)の活用は、大手企業では進み大きな効果を挙げているが、多数存在する中小のデザイン事務所ではあまり進んでおらず、デザイン業界全体で見ると、3次元CGはまだ限られた利用にとどまっている。その理由は、3次元CGソフトウェアの使いこなしの難しさと、高価な設備投資にあり、人的、資金的に余力のある大手企業のみが、3次元CGの効果を享受できる状況にある。

2. 研究の目的

このような状況を打開し、デザイン分野における3次元CG活用の裾野を広げ、業界全体のデザインの質の向上や競争力の向上を図るために、デザイン検討をターゲットとし使いやすさに重点を置いた3次元CGのソフトウェアを開発し、そのソフトウェアをインターネットを介して公開して、産業界に普及展開する。

3. 研究の方法

大別して、「CGレンダリングソフトウェアの研究開発」と、「開発ソフトウェアの普及展開の仕組み構築」の二つについて実施した。

CGレンダリングソフトウェアの研究開発では、CG表現の高品質化、使いやすさの向上、計算時間高速化の研究に重点を置き、すでに保持しているプロトタイプソフトウェアの改良によって性能を向上させると共に、デザイナーにとって有用なツールとなるための実用性を追求した。

開発ソフトウェアの普及展開の仕組み構築では、本ソフトウェアを紹介するホームページを立ち上げ、デザイナーがインターネットを介してソフトウェアをダウンロードできるようにした。また、あわせてユーザーマニュアルの整備を行った。

これらの作業のうち、プログラムのコーディング、ホームページ立ち上げ、マニュアル作成には学生アルバイトによる作業を活用した(延べ1370時間)。

4. 研究成果

(1) CGレンダリングソフトウェアの開発

建築デザインに特化し、高機能と使いやすさを両立させた独自のCGレンダリングソフトウェア「Frend」を完成させた(図1)。

本研究で開発したソフトウェアの機能の一覧は、以下である。また、表1に開発ソフトウェアの関数の一覧を示す。ソースコードは約50,000行の規模になった。

材質に関する機能

拡散反射物質、透過物質、鏡面反射物質、屈折物質、簡易鏡面反射、ハイライト、材質表面の模様、材質表面の凹凸、マッピングのエッジブレンド、鉛直マッピング、ラップマッピング

光源に関する機能

点光源、スポット光源、線光源、環境光源、光源の距離減衰の調整、太陽位置指定、陰部分の立体感表現、グローライト、影の表現、間接光シミュレーション

構図に関する機能

透視投影、平行投影、強制2点透視投影、視野角、スクリーン設定による断面表示

特殊効果の機能

背景画像の合成、全天空への空画像の配置、フォグによる遠近感表現、レンズフォーカスによる遠近感表現、線画表現の混在、パノラマインサイド、パノラマアウトサイド

アニメーション機能

視点の動き、物体の動き、光源の動き、フェードインフェードアウト、映像のマッピング、背景の変化、スクリーン位置の変化、視点経路や速度の局所コントロール、フィールドレンダリング

作画機能

アンティエイリアシング、部分レンダリング、デュアルプロセッシング、バッチレンダリング、ネットワークレンダリング

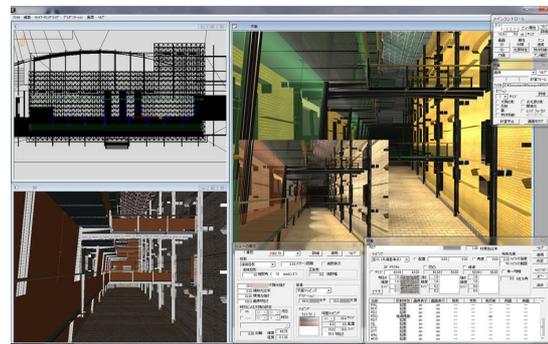


図1 開発ソフトウェア「Frend」の操作画面

このソフトウェアを実現させる過程で研究開発した機能のうち、代表的なものについて記す。

(2) 間接光シミュレーション機能の開発

レンダリングの計算時間は、特に相互反射を考慮した間接光計算での負荷が大きい。通常のCGでは、光源から物体に直接到達する直接光成分(図2)のみを扱うが、特に建築分野でのデザイン検討への適用では、実世界をシミュレーションするために、光源からの光が周囲の物体で反射した後で到達する間接光成分(図3)も含めて計算することが不可欠である。しかし、間接光計算は、図3に示すように計算の複雑さとそれに伴う計算時間の負荷が膨大である。そこで、CGをデザイン検討ツールとして活用するために、間接光計算を高速に処理するアルゴリズムを研究開発した。図4は従来の直接光成分のみの計算による画像例であり、本研究で開発した間接光成分も含めた計算アルゴリズムで計算した結果が図5である。図4と図5の画質の違いが示すように本研究により大幅なリアリティの向上を実現させた。

表1 開発ソフトウェアの関数一覧

関数名	説明
aloc_animg	アニメーション画像の領域確保
aloc_preindari	間接光、面光源の領域確保
aloc_stimg	静止画像の領域確保
antia	アンチエイリアス処理
antia_jdg	アンチエイリアス判定
antia0	アンチエイリアスなしの処理
cdifaf	材質の反射特性コード変換
depefc	デプスマップ(モヤ)効果
extarl	面光源の抽出
extmap	マッピングファイル名の種類を抽出
extmapnm	マッピングファイル名の抽出
chmsz	マッピング画像のサイズによって画像を分類
fldrm	フィルターレンダリング
flsim	光源の配光経路方向のデータをシメトリ展開
focus	焦点マウス処理
free_tmtx	マッピング変換情報の解放
free_pers	透視変換情報の解放
free_parsbl	透視変換情報の解放 (ホール用)
free_once	静止画像・アニメーション情報の解放
frmct	点対面のフォームファクタの計算
frmrd	フレームレンダリング処理
inouts	パンチングボリュームでの内外判定
intrhl	穴の交点計算
invmtr	逆行列の計算
ktbpxl	ボクセルとレイの交点計算
mkpnt	配光データのグリッド点での値を計算
normal	ポリゴン・CSGプリミティブの法線ベクトルを得る
normtr	透明ポリゴンの法線ベクトルを得る
nxtblx	次のボクセルの検索
direct	面光源の直接光成分の計算
arhnt_ar	面光源のフォームファクタと照度計算
aromad	面光源の重要度サンプリング
avrcol	色の平均値計算
idacnv	変化を使用した集合判定
pntsm	ポリゴンのパッチ番号の重要度サンプリング
pntsm	パッチにおけるポイントの様なサンプリング
mtrarl	面光源のワールド座標→平面座標の変換行列の計算
mtrdir	平面座標→ワールド座標の変換行列の計算
occlusion	イベントオクルージョン処理
indirect	間接照明処理
susmp0	シミュレーションフォームサンプリング処理(初回)
indref	拡散反射の反射方向
diffuse_id	拡散反射計算(間接光用)
srhlst_id	太陽の拡散反射計算(間接光用)
nrhlst_id	点光源の拡散反射計算(間接光用)
srhlst_id	サンプリングによる影計算(間接光用)
avrcol_id	色の平均値計算(間接光用)
susmp	シミュレーションフォームサンプリング処理
free_infr	間接光計算のデータ解放
invpdf	重要度サンプリングのための分配関数
randm1	乱数取得(一様)
randm2	乱数取得(Sin x Cos)
scanln_path	間接光計算用のスキャンライン処理(間接光反射物体のみ)
scanln_path_on	間接光計算用のスキャンライン処理(すべての物体)
omtcol	ポリゴンのメッシュポイントの計算省略
ipomcl	同じポリゴンのメッシュポイントの計算補充
putavr	未計算メッシュの色の平均値を出力
indfilter	間接光計算の画像ぼかし処理
intrpl	画像の補間処理
ind_off	間接計算のnd効化
ind_off2	指定した座標の間接光計算無効化
samepl	指定した座標が同一ポリゴンに含まれるか判定
ndcol	ポリゴンの色の平均値計算
raytrc	間接光用レイトレーシング 初期追跡
trace0_p	間接光用レイトレーシング 初期追跡
nttblx	ポイントの属するボクセルを求める
prebview	物体の動きに連動した視点アニメの前処理
premov	BREPSと光源を交互にシミュレーション
prepb	plane-set bounding volumeのプリ計算
psbvt	plane-set bounding volumeによる交点判定
aloc_brp	BREPSのデータ領域確保
cnvlay	レイを平面からコードに変換
deflbt	面の左下座標・中心座標の定義
defnrm	法線ベクトルの作成
defvol	plane-set bounding volumeの作成
extmtr	BREPSのマテリアル名の抽出
free_brp	BREPSのデータ解放
psbrp	BREPSのポスト処理
rdbrp	BREPSデータをファイルから取得
rdosg	CSGデータをファイルから取得
rdidt	データ用ディレクトリデータをファイルから取得(未使用)
rdicvt	スムースシェーディングのためのベクトルデータをファイルから取得
psfrm	フレームレンダリングポスト処理
rdfrm	フレームデータをファイルから取得
rdkeyfrmall	キーフレームデータをファイルから取得
wkeyfrmall	キーフレームデータをファイルに書き込
mk_gwpl	グローバルポリゴンの作成
psltht	光源データのポスト処理
rdlht	点光源・線光源のデータをファイルから取得
rdltbl	配光特性データをファイルから取得
rdlumi	配光データの取り出し
type1	type1の配光情報を取得
type2	type2の配光情報を取得
type3	type3の配光情報を取得
type4	type4の配光情報を取得
wltht	光源データをファイルに書き込
intrmtr	マテリアルパラメータの初期化
psmtr	マテリアルデータのポスト処理
psrdmap	マッピングデータ読込のポスト処理
scmap	マッピングイメージのスケール処理
micnv0	オリジナルマップからミップフォーマットに変換
rdmtr	マテリアルデータをファイルから取得
rdmxt	テクスチャデータをファイルから取得
rdmpt	パンプデータをファイルから取得
rdopc	透過パターンデータをファイルから作成
wmtrt	マテリアルデータをファイルに書き込
ifefcd	マテリアルデータのコード変換
getvis	5桁のコードを5つのコードに分割
rdmvo	物体の動きのデータをファイルから取得
wtmvo	物体の動きのデータをファイルに書き込
diffuse	拡散反射処理
seelht	視点方向からの光源より明るさ計算
ambien	環境の明るさの加算処理(太陽光)
ambien2	環境の明るさの加算処理
arhnt	面光源の拡散反射計算
diffuls	発光する面の処理
hlit	ハイライト処理
indrc	間接光計算
lambprt	ランベルトの法則
lnlht	線光源の拡散反射計算
lnlnt	点光源の光計算
noshde	陰なしの拡散反射計算
ptlht	点光源の拡散反射計算
ptlal	点光源の拡散反射計算(全て)
ptlstm	点光源の拡散反射計算(サンプリング)
srhlst	全光源の拡散反射計算
amax1	3つの指定値の中から最大値を取得
cnvrb	色相からRGB値に変換
distmt	距離を指定した単位からメートルに変換
distmt_inv	距離をメートルから指定した単位に変換
get_netname	ファイルパスからネットワーク名を取得
getdir	ファイルパスからディレクトリ名を取得
hlrgb	HLS色空間からRGB値に変換
inoubx	点がボックスの内か外の判定処理
inoutf	点がCSGプリミティブの内か外の判定処理
inters	レイとCSGプリミティブの交点を取得
igovpo	凸ポリゴンか否かの判定処理
igovlo	点がポリゴンの境界の中心ノ外にあるかの判定処理
ktbpx	ボクセルとレイの交点を取得
ktbpl	プレーンとレイの交点を取得
ktmnel	楕円体とレイの交点を取得(最小)

ktmxel	楕円体とレイの交点を取得(最大)
mkvct	2点から単位ベクトルを取得
nrmel	楕円体の法線ベクトルを取得
nrmvct	ポリゴンの法線ベクトルを取得
plcut	面の方程式を取得
prjcol	3次元空間のポリゴンを2次元平面に投影
rgblht	RGB値から明度を取得
set_fullname	プロジェクト相対パスから絶対パスに変換
set_relativenam	絶対パスからプロジェクト相対パスに変換
sundelt	太陽赤緯・均時差の取得
sunvct	日時から太陽方位を取得
tmnc3	2つの行列の乗算処理
tmcp3	4x4の行列コピー
tmint3	4x4の単位行列作成
tmrtx3	アフィン変換行列取得(X軸回転用)
tmrty3	アフィン変換行列取得(Y軸回転用)
tmrtz3	アフィン変換行列取得(Z軸回転用)
tmrsl3	アフィン変換行列取得(スケール用)
tmrtr3	アフィン変換行列取得(平行移動用)
trnsf3	アフィン変換行列によるベクトル変換(3x4)
trnsf4	アフィン変換行列によるベクトル変換(4x4)
trnsn3	アフィン変換行列のコピー(平行移動用)
txpl	テクスチャ座標の取得(未使用)
untvct	3点から単位ベクトルを取得
vctang	2つのベクトル間の角度を取得
vspro	2つのベクトルの内積を取得
bmpmap	パンプマッピング処理
bpel	楕円体のパンプマッピング処理
bppl	面のパンプマッピング処理
edgblid	マッピングのエッジの混合処理
maping	マッピング処理
mapmtr	マッピング時の変換行列を取得
mappix	ポリゴン座標からマッピング座標に変換
jdg_mapsmpl	マッピング時のサンプリング方法の判定
set_mapsmpl	スーパーサンプリングを行う、マッピング座標の取得
mipmap_pnt	ポイントサンプリングでミップマップ作成
mipmap_spr	スーパーサンプリングでミップマップ作成
mipmapbmp	ミップマップから法線ベクトルを取得(パンプマッピング用)
opcmab	不透明マッピング処理
opcmab_bil3d	不透明マッピング処理(3Dビルボード用)
premi	ミップマップパラメータの取得
scr2wrd	スクリーン座標→ワールド座標の変換
txtmap	テクスチャマッピング処理
txtmap_wrap	テクスチャマッピング処理(ラップ)
txtmap_bil3d	テクスチャマッピング処理(ビルボード用)
movbrp	物体の動きの処理
brrot	変換行列を生成し、任意の軸でBREPSを回転
brcsl	視点の動きを設定、任意の軸でBREPSをスケール
brmtr	変換行列を生成し、任意の軸でBREPSを平行移動
bview	最初のフレームビューと現在のフレームビューから変換行列を取得
fade	フェードイン・フェードアウト処理
mtrtx	ワールド座標→平面座標の変換行列の取得
mtrtxc	ワールド座標→平面座標の変換行列の取得(CSGプリミティブ用)
mtrtxd	ベクトルを(1, 0, 0)に変換(未使用)
glwscut	直接影の取得
mtrtxwr_gl	OpenGL ランプ投影用の変換行列を取得
btmnsf	物体の動き
btmnsf_bil	ビルボード用にBREPSの座標変換を行う
ptrnsf_glbil	グローバルポリゴン用にBREPSの座標変換を行う
movint	物体の動き操作の初期化処理
mvbrcx	物体の動きの実行
mvmtex	材質の動きの実行
ptmnsf	点光源の動き
var	視点の動きを設定(未使用)
aloc_pershl	ホールの透視変換の領域確保
aloc_perspl	ポリゴンの透視変換の領域確保
aloc_tmtx	マッピング変換行列の領域確保
defmtr	マッピング変換行列の取得
hlpers	ビュー座標→ハースベクトル座標変換(ホール)
mvfr	ルをスクリーンでカットする
mtrsc	ローカル座標→ワールド座標の変換行列を取得
plmnm	ポリゴンの最小値・最大値を取得
plcut	視点から裏向きに見えるポリゴンのカット
plpers	ビュー座標→ハースベクトル座標変換(ポリゴン)
plscut	ポリゴンをスクリーンでカットする
rdwscut	グローバルポリゴンをスクリーンでカットする
prescn	スキャンライン処理の前処理
wndcut	ウィンドウ外のポリゴンをカットする
foerr	ファイル名に関連したメッセージ出力
getf	文字列から数値を取得
geti	文字列から整数を取得
rdline	ファイルから行読み込みカンマで分割したデータを取得
devline	文字列をカンマで分割する
rd_alldata	全てのデータをファイルから取得
rdfilename	全てのファイル名を取得
rdsky	全てのデータを取得(ビットマップ形式)
rd_arimg	面光源のデータを取得(ビットマップ形式)
rd_indimz	間接成分の計算結果の画像を読み込み
rdspd	動きの速度データをファイルから取得
cnccdg	面のエッジと穴のエッジの合成
cnccdgs	面のエッジと穴のエッジの合成(スムージング面用)
ispap	面をスキャンラインでスライスしてエッジを抽出
ispaps	面をスキャンラインでスライスしてエッジを抽出(スムージング面用)
scanln	スキャンライン法による隠面消去
scantr	スキャンライン法による隠面消去(透過面用)
scnfl	ピクセルを面で塗りつぶす
scnfls	ピクセルを面で塗りつぶす(スムージング面用)
scnft	ピクセルを面で塗りつぶす(透過面用)
scnfst	ピクセルを面で塗りつぶす(スムージングの透過面用)
srtrt	透過面の場合にZ値でソートする
srtrts	透過面の場合にZ値でソートする(スムージング面用)
mkbmpname_an	ビットマップファイルを作成(アニメーション用)
wtbmo	ビットマップファイルの書き込
wtbmphp	ビットマップファイルのヘッダ部の書き込
rdmnd	樹木の表現
refct	鏡面反射計算
refvct	反射法線計算
rflcoe	透明オブジェクトの反射係数計算
rfrcoe	屈折率の計算
rflnt	点と線の関係を得る
scnwr	1スキャンラインの書き込
callf	レンズフォーカスのためのフィルタの計算
shade	陰計算
shadow	影計算
shadwc	CSGプリミティブの影計算
sky	背景処理
skymap	空の画像の球面マッピング処理
skymapl	空の画像の平面マッピング処理
pxblid	画像のエッジのブレンド処理
smpsdw	サンプリングによる影計算
trace	レイのトレーシング
trace0	初回レイトレーシング
trace2	2回以降レイトレーシング
trace3	CSGプリミティブのレイトレーシング
dspranz	輝度をディスプレイの明るさに変換
trngb	RGB値の移し替え
trnsee	視点を視点座標系からワールド座標に変換
trnseef	視点を視点座標系からワールド座標に変換(魚眼レンズ用)
tmnsf	透明オブジェクトによる透過と反射に分離

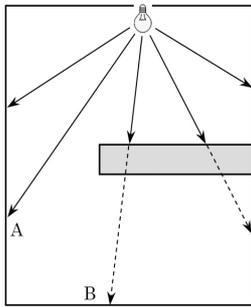


図 2 直接光成分

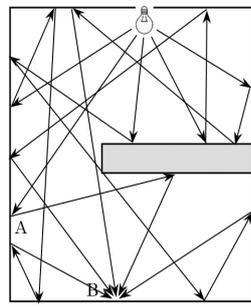


図 3 間接光成分

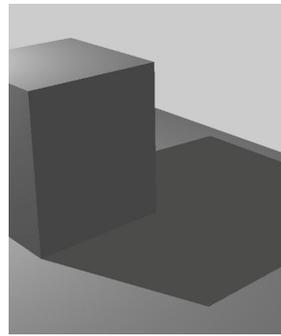


図 6 従来の影

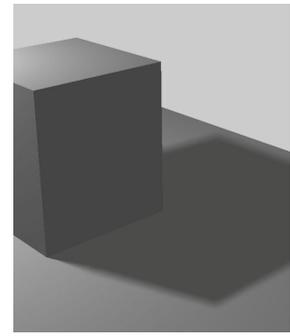


図 7 本研究による影



図 4 直接光による従来の計算例



図 5 本研究による計算例

(3) 点光源による半影表現の研究開発

点光源では、影である部分と影でない部分が明確に区別されるのに対し、面光源は、光源からの光が全て到達する部分と完全に遮られる部分との間に、面光源の光の一部だけ到達する部分（半影）が生じる。半影が加味された面光源計算の画像の方がより現実に近いのは明らかであるが、面光源の影を計算するには膨大な計算時間が必要となる。そこで、計算高速化のために、面光源を点光源で近似して計算するアルゴリズムを研究開発した。これによって、点光源の高速な影計算時間で、半影によるリアルな影表現を実現した。図 6 は 点光源による従来の影表現の例、図 7 は開発アルゴリズムによる点光源の半影

表現の例である。同程度の計算時間で、より自然な影の表現を実現させた。

(4) GPGPU による高速化検討

CG のレンダリング計算技術は、近年ますます高度化し、それにともない計算時間が増大していることが、デザイン段階での CG 利用のネックになっている。近年は CPU の計算性能の向上があまり期待できなくなっている状況の中で、これまで特殊な用途のためのものであった GPU が汎用的な計算処理に利用できる道が開けてきた。計算時間のかかる CG 計算処理を GPU で行うことができれば、計算時間が飛躍的に向上し、設計プロセスの効率化が期待できる。そこで、GPU で計算するにはどのようにコーディングするのか、GPU 化によって高速化がどの程度期待できるのかを検討した。

GPU の演算資源を汎用計算に応用する技術である GPGPU の開発環境として、OpenCL と CUDA があり、本研究では CUDA を採用した。

レンダリング機能を鏡面反射と影だけに制約した基本的レイトレーシングによる計算時間測定用テストプログラムを作成して、GPGPU 化の効果を検討した。図 8 が計算時間測定のために作成したベンチマーク画像で、表 2 がその測定結果である。測定に使用した GPU は、NVIDIA GeForce の GTX580、比較対象の CPU は、Core i7-2630QM 2.0GHz を使用した。

表 2 を見ると CPU 計算と GPU 計算の計算時間の差は顕著で、1024 x 768 サイズで 94.7 倍の高速化を達成している。特に、計算時間がかかるものほど高速化率が高いことがわかり、GPGPU 化のもつ大きな可能性を明らかにした。ただしこの結果はプロトタイププログラムによる計算測定であり、レンダリングのフル機能を装備したプログラムで同程度の速度を実現させるには、今後更なる検討が必要である。



図 8 ベンチマーク画像

表 2 CPU と GPU の計算時間比較

画像サイズ	CPU 計算 (i7-2630QM)	GPU 計算 (GTX580)	速度比
256 × 192	2.7 sec	0.1 sec	33.1
512 × 384	10.5 sec	0.2 sec	69.9
1024 × 768	41.7 sec	0.4 sec	94.7

(5) 開発ソフトウェアの普及展開

本ソフトウェアを公開するホームページを開設し、デザイナーがインターネットを介してソフトウェアをダウンロードできる仕組みを構築した(図9)。またあわせて、ユーザビリティ向上のため約150ページの操作マニュアルの整備を行った(図10)。

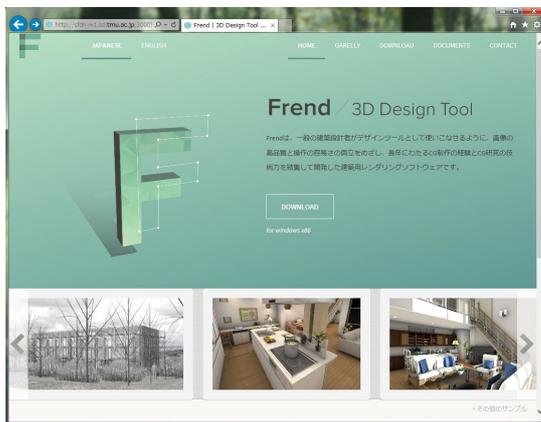


図9 Frendの公開サイト

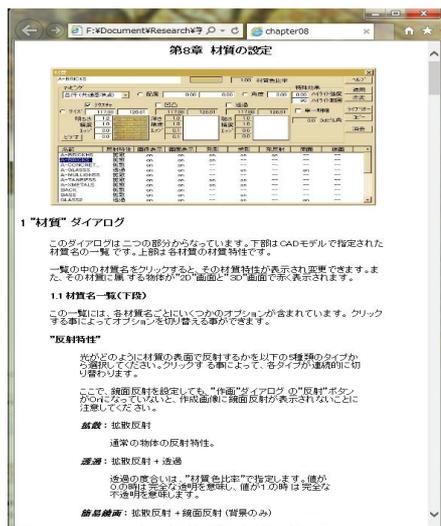


図10 操作マニュアル

以上、高品質と使いやすさの両立を目指したCGレンダリングソフトウェアを研究開発し、「Frend」ソフトウェアを実現させると共に、インターネットを介して開発ソフトウェアを普及展開する仕組み構築を実現させた。今後は本ホームページを通して本開発ソフトウェアを公開して普及展開し、産業界への貢献を図ると共に、ユーザーからのフィードバックを受けて、さらにユーザーの要望を

反映させたソフトウェアの改良を進める。また、本ソフトウェアの有効性を検証するために、使用するCGソフトウェアの性能の違いがデザイン作業の品質や効率に同様に影響するについても今後検討を進める。

5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

〔雑誌論文〕(計1件)

- (1) 丁慶松、高碩、笠原信一、「GPUを活用したCGレンダリング超高速処理システム開発のための予備実験 その1-GPUによる高速処理効果の予測-」、審査付き研究報告採用報告、日本建築学会、2012年6月、ISSN 1881-512X

〔学会発表〕(計5件)

- (1) 吉村ももこ、笠原信一、安藤大地、「レイトレーシングにおける半影表現の新しいアプローチ」、情報処理学会グラフィックとCAD研究会第154回研究発表会、2014年2月20日、理化学研究所和光本所(埼玉県和光市)
- (2) 黒田宙馬、笠原信一、安藤大地、「3Dインターフェースによるプレゼンテーションツールの可能性提案」、情報処理学会インタラクション2014、2014年2月28日、日本科学未来館(東京都)
- (3) 多々良友理、笠原信一、安藤大地、「モザイク画像の認知に関する考察とそれに基づいたグラフィックデザイン支援プログラムの開発」、情報処理学会インタラクション2014、2014年2月27日、日本科学未来館(東京都)
- (4) 丁慶松、笠原信一、「GPUを活用したCGレンダリング超高速処理システム開発のための予備実験 その2 並列処理最適化アルゴリズムの考察」、日本建築学会第83回関東支部研究発表会、2013年3月7日、建築会館(東京都)
- (5) 高碩、笠原信一、「GPUを活用したCGレンダリング超高速処理システム開発 その3 シングルGPUからマルチGPUへの拡張検討」、日本建築学会第83回関東支部研究発表会、2013年3月7日、建築会館(東京都)

〔ホームページ〕

- (1) 開発ソフトウェアFrendの紹介サイト <http://sfdn-w1.sd.tmu.ac.jp:30005>
- (2) 首都大学東京笠原研究室のサイト <http://sfdn.sd.tmu.ac.jp/>

6. 研究組織

(1) 研究代表者

笠原 信一 (KASAHARA SHINICHI)
 首都大学東京・システムデザイン学部・教授
 研究者番号：00433178