物理ベース?アート指向?

~尤もらしさと自由度を両立するレンズフレア表現~

シリコンスタジオ株式会社 研究開発室室長兼 ミドルウェア技術部フェロー 川瀬 正樹

masa@siliconstudio.co.jp



- レンズフレア
- レガシーな手法について
- <u>物理ベースレンズフレアについて</u>
- 既存技術の課題
- 尤もらしさと自由度の両立を目指して
- <u>実装のための光学効果等の詳細</u>
- 生成結果サンプル
- その他実装に関するノウハウ
- <u>まとめと今後の展望</u>



フレアとゴースト

- ・フレア
 - 眩しい光の迷光により白っぽくコントラストが下がる現象
 - 主に絞りやフィルタによる回折で明るい光が広がる現象
 - ・ブルーム
 - 光条(スターバースト)
- ゴースト
 - 光学系(レンズ)の内部反射により発生する虚像が並ぶ現象
- 総称してよくレンズフレアと呼ばれる

フレアとゴースト





フレアとゴースト



enced from [カメラの、うろかいなのいう。フレア・コーストの内正方法。] <u>Wise Camera https://camera-beginner.sakura.ne.jp/wp/?p=669</u>

ゴーストの発生メカニズム



Figure 9: One out of 91 possible double-reflection sequences for this lens design

Referenced from "Physically-Based Real-Time Lens Flare Rendering" [Hullin et al. 2011] http://resources.mpi-inf.mpg.de/lensflareRendering/

レガシーな手法について

スプライトベースのレンズフレア

- 光源から画面中心の点対称に虚像のスプライトを並べて描画
- 昔からよく使われてきた手法
- 自由度が高い
- クォリティが高い(潜在的に)
- 点光源を前提
- 光源数に比例して負荷が増大
 - 事実上小数の点光源にしか使えない
- 反射光等には原則として使えない
 - パフォーマンスが充分であればスキャッターベースで可能
- モーションブラーを適用できない
 - レンズフレアだけにブラーがかからず不自然に見えることがある

アーティストがゴーストを定義

- ゴーストの形状や位置を定義
 アーティストやプログラマが記述
- 実行時は光源位置を画面中心を基準にスケール



©2005 BUNKASHA PUBLISHING CO.,LTD. ©2005 Microsoft Corporation. All rights reserved.

イメージベースのレンズフレア

- ポストプロセスの画像処理
- 再帰的に画像全体を点対称スケーリング [Kawase 2003]
- 自由度は高くない
- クォリティはあまり高くない
- 光源の数に関わらず負荷が固定
 - 光源が何千何万あっても変わらない - 逆に一切無くても固定の負荷がかかる
- 面光源や反射光からも生成可能
 モーションブラーにも対応可能



Image Based Lens Flares [Mittring 2012]

Source Image with Bloom

IB Lens Flares (without Lens Blur)



Lens Blur Sprite Image Silicon Studio Corp., all rights reserved. IB Lens Flares (with Lens Blur)

IB Lens Flares Examples [Mittring 2012]



Emissive (Sun)

Emissive (Fire)

Reflections

物理ベースレンズフレアについて

リアルタイム向けの実装

- レンズの特性を利用して事前計算
 光学系(レンズ)は原則として同心円(等方性)
 - アナモルフィックレンズ
 - 横方向のみの挙動になるだけなので等方とみなせる
 - ティルト/シフトレンズ
 - 中心をずらす事で等方になる
 - ・絞り羽根
 - 光束の軌跡は等方(絞りは単なるマスクとみなせる)
 - 光束の挙動は光軸からの高さと角度だけで決まる
 - レンズ中心からの同心円上の距離と入射角度

描画候補となるゴースト

- ・屈折面は光学系を構成するレンズの2倍

 レンズエレメント毎に入射と出射
- ゴーストは必ず偶数回の反射結果 [Hullin et al. 2011]
 2回反射や4回反射の結果がゴースト候補となる

描画候補となるゴースト

- ほとんどの光線は透過
 - AR(反射防止)コーティング(後述)
 - 屈折面毎の反射率はおおよそ1%未満
- 4回以上の反射はほぼ見えない
 - 各々1%の反射なら2回反射で0.01%(1万分の1)
 - 4回反射なら0.00001%(1億分の1)
- 事実上2回反射の結果のみがゴーストとなる

近年のリアルタイム物理ベース手法

- Physically-Based Real-Time Lens Flare Rendering, SIGGRAPH 2011 [Hullin et al. 2011]
- Practical Real-Time Lens-Flare Rendering, EGSR 2013 [Lee et al. 2013]

Physically-Based Real-Time Lens Flare Rendering [Hullin et al. 2011]

- http://resources.mpi-inf.mpg.de/lensflareRendering/
- 事前計算情報から128x128グリッドでゴースト描画
 グリッド間はGPUによる補間
- 画質は非常に高い
 - ゴーストの複雑な変形や収差表現
 - コースティクス(集光模様)
- 事前計算情報が多く実行時負荷も高め
 - 例えば事前計算
 - ゴースト毎に90方向×642レイ×20ズーム係数と8絞り
 - 実行時
 - 128x128グリッドで描画

Practical Real-Time Lens-Flare Rendering [Lee et al. 2013]

- 光線行列を利用した変換(Matrix Optics)でゴースト描画
 - 光線の高さと入射角情報を行列変換して4x4グリッドを描画
 - 事前計算がほぼ不要で実行時も非常に高速
- 近軸理論(Parallax Optics)
 - $sin(\theta) = \theta, cos(\theta) = 1, tan(\theta) = \theta$ による近似
 - 光線の屈折と反射による挙動を線形変換行列で表せる
 - 複数枚のレンズをまとめて行列表現
- 線形変換が前提なので非線形な効果は表現できない
 - ゴーストの複雑な変形
 - コースティクス
 - 単色収差
 - 単色収差は近軸理論と正確な計算との差なので原理的に発生しない
 - 色収差は線形変換でも表現可能

既存技術の課題

レガシーな手法の課題

- 制作するアーティストへの依存が大きい
 - アーティストの制作負荷が高い
 - クォリティが製作者依存となる
 - 尤もらしい表現には専門的な知識や経験が必要
- 複雑な形状や挙動のゴーストを作ることが難しい
 - ズームや絞り変化への適応が難しい
 - コースティクスやケラレの変化など

物理ベース手法の課題

- レンズデータが必須
 - ズーム等の変化も必要
- 物理的制約に縛られやすい
 - 単焦点レンズではズーム出来ない
 - ズーム域が足りない(ある程度対処は可能:後述)
- カスタマイズが難しい
- アルゴリズムによっては負荷が高い
 - 事前に用意しておくデータ量が多い
 - 実行時負荷が高い
- 高速な手法では表現力がやや落ちる
 - 例えば線形変換のみでは複雑な変形やコースティクスを表現できない

尤もらしさと自由度の両立を目指して

コンセプト

- 手軽にレンズフレアを制作できる
 アーティストが自由にカスタマイズ可能
- ズームや絞りの状態に適応した挙動
- 物理的正確性には拘らない
 - 実用上尤もらしく見えれば充分
- 複雑な挙動や形状のゴースト表現
 - 複雑な変形
 - コースティクスやハードエッジ
 - 鏡筒によるマスク(ケラレ)

基本的な描画方法

レガシーなスプライトベースゴースト

- ゴースト毎に位置やサイズ情報を定義
- 大半の重要でないゴーストは乱数で生成
- 光学的な影響を考慮
 - 絞り、ズーム、分散、輝度やサイズの変化

– etc.

実装のための光学効果等の詳細

考慮すべき光学的効果

- 反射防止膜(ARコーティング)
- 絞り羽根表現
- ゴーストの位置とサイズ
- ズーミングによる変化
- ゴーストの輝度
- 鏡筒によるマスク(ケラレ)
- 分散/色収差
- 非線形な変形
- 回折フレア

反射防止膜(ARコーティング)

レンズの反射率

- 一般的なガラスは屈折率1.5前後
 - 正面からの入射を4%程度反射
 - 斜めからの入射光はより強く反射
- 光学系はおおよそ10枚単位のレンズで構成される
 - 屈折面はレンズ枚数の2倍
 - 光学系全体では数十%が反射で失われる

反射防止膜 (ARコーティング: Anti-Reflective Coating)

- ・反射防止の主流の手法
 - 薄膜のコートで反射光を干渉させて打ち消す
 - 薄膜でレンズ本体と位相が180度ずれた反射を起こして干渉
 - 単層から複数層コーティングまで
 - 多い程多くの波長で強く反射を抑えられる
 - 波長と入射角によってはほとんど反射ゼロに
- 反射防止膜による着色
 - 反射光は打ち消せなかった波長で着色される
 - 反射率はコーティングの種類と入射角で大きく異なる

単層反射防止膜の波長と入射角による反射スペクトルと 実際のレンズの反射の様子 [Hullin et al. 2011]



Figure 4: Lens coatings make reflections from optical interfaces inside the lens barrel appear colored. Left: Canon EF 70-200mm f/2.8L. Middle: Canon EF 100mm f/2.8 USM macro. Right: Net reflectivity in % of a quarter-wave coating designed for 532 nm light at normal incidence ($n_{glass} = 1.5$, $n_{coating} = 1.38$, d = 96.4 nm).

ゴーストの着色

- ・全て2回反射と仮定
 - 2種類のコーティングによる反射色の乗算
- レンズによって残存反射の色を何種類か用意する
- ゴースト毎に2種類の色を選択して乗算
 - 特定の2種類のコーティングで反射したものとみなす
 - 本来はレンズとコーティングの順序を定義する必要あり

• 実用上は乱数でもほとんど問題ない



ゴーストの多くは絞り形状でマスクされる



Figure 9: One out of 91 possible double-reflection sequences for this lens design

Referenced from "Physically-Based Real-Time Lens Flare Rendering" [Hullin et al. 2011] http://resources.mpi-inf.mpg.de/lensflareRendering/

Distance Field (距離場)

- 絞り羽根形状を表現する距離場から生成
 - 距離場を補間することで絞り状態の滑らかな遷移が可能
 - 色収差や非線形な変形の表現にも応用できて便利(後述)
- ・距離場の生成
 - 事前生成テクスチャ
 - ランタイムプロシージャル


距離場の作成

• 事前計算テクスチャ

- ボケの絞り形状の生成に使用 [Kawase 2012, 2015.a]
 - そのまま流用できる
- 事前生成により非常に複雑な形状が可能
 - 絞り羽根の微細な歪み表現など
- 質と負荷がテクスチャ解像度で決まる



事前計算距離場マップ(DOF絞り形状生成用)

少し絞った状態 絞り込んだ状態





開放絞り

7枚絞り

事前計算距離場マップから生成した絞り形状



距離場の作成

- ランタイムでのプロシージャル生成
 - 座標を引数として多角形や円形絞り形状を生成
 - 事前生成データが不要
 - 解像度に依存しない
 - あまり複雑な形状は表現できない
 - 負荷が高くなり現実的ではない

ランタイムでプロシージャル生成

- 今回は実行時にプロシージャル生成
 - 微細な歪み等はゴーストではあまり重要ではないため
 - 必ずしもDOFの絞り形状に縛られない表現(自由度)のため - N角形を生成する汎用コードを書いて特殊化すると良い
- •参考:
 - 2D distance functions

http://iquilezles.org/www/articles/distfunctions2d/distfunctions2d.htm

7枚絞りの距離場生成例

const int nSides = 7; const float PI = 3.14159265359f; const float fTheta = PI / nSides; const float fCosTheta = cos(fTheta); const float fSinTheta = sin(fTheta);

const float fTheta2 = 2.0f * fTheta; const float2 k2 = float2(cos(fTheta2), sin(fTheta2)); vPosition.x = abs(vPosition.x); vPosition -= 2.0f * min(dot(k2, vPosition), 0.0f) * k2; vPosition -= 2.0f * min(dot(float2(-fCosTheta, fSinTheta), vPosition), 0.0f) * float2(-fCosTheta, fSinTheta); vPosition -= 2.0f * min(dot(float2(fCosTheta, fSinTheta), vPosition), 0.0f) * float2(fCosTheta, fSinTheta);

const float r = fRadius * fCosTheta; // 内接円の半径 const float fTanTheta = tan(fTheta); float d = length(vPosition - float2(clamp(vPosition.x, -r*fTanTheta, r*fTanTheta), r)); if (vPosition.y < r) d *= -1.0f; const float r0 = length(vPosition) / fRadius; d = 1.0f + d / r; d = lerp(d, r0, fCircularity); // 外周を多角形から円形に滑らかに補間

距離場から生成する絞り形状の制御

- 絞りの円形度に応じて円形と多角形の距離場を補間
- エッジ部分は滑らかに減衰させる
 - アンチエイリアスとしての効果もあり

プロシージャル生成した距離場と絞り形状



ゴーストの位置とサイズ

ゴーストの位置

- 画面中心と光源を結ぶ直線状に並ぶ
 - 画面中心から光源に向かうベクトルを定数倍した位置
 - 画面中心を原点とする座標系

• g = sl

g:ゴーストの座標

1: 画面中心から光源に向かうベクトル

s:ゴースト毎に定義されたスケーラ(±2~4程度の範囲)

- 一様に並ぶ訳ではない
 - 基本的にはsは乱数で良い
 - 制御したい小数のゴーストのみ手動で定義
 - ある程度偏りがある
 - 光源の点対称側(s = -1 付近)はゴーストが集まりやすい

间面中

ゴースト位置のばらつき

- レンズは理論上は完全に等方
 実際のゴーストの位置はばらつきがある
- ゴースト毎に乱数で原点をずらす
 - 一様な乱数では不自然
 - 大きくずれるのは極一部
 - 乱数をべき乗するなどして偏りを表現



ゴースト位置のばらつき <u>なし</u>



ゴースト位置のばらつき あり



ゴースト位置の非線形性

- 光源の画面中心からの距離による変化 [Kawase 2012]
 - レンズは等方性だが非線形のカーブをもつ
 - ゴーストの位置も非線形で複雑なカーブを描く
- 実装
 - sを光源ベクトルlの長さで変化
 - 例えば|1|をべき乗した値でsをスケール
 - ゴースト毎に固有のパラメタを与えて挙動を変化させる

ゴーストのサイズ

- 被写界深度の錯乱円径(Circle of Confusion: CoC)に相当
 - ただし明確なフォーカスは存在しない
 - より複雑な挙動
- レンズの設計によって様々な個性がある
 - 正確な表現はシミュレーションが必要
 - 実用的には殆ど乱数で良い
 - 制御したい小数のゴーストのみ手動で定義
 - 絞りや画角によって光学に基づいて動的にスケール

ゴーストのサイズ

- 絞りにしたがって変化
 - 絞り有効径か実効F値の逆数に比例させる
 - センササイズや開放絞り値の影響を受けるかどうか?
 - 光学的にはセンササイズは影響し、開放絞りは影響しない
 - ただし何の影響を受けると使いやすいかはケースバイケース
 センサが小さくなるとゴーストも小さくしたい?
- 絞りを通るタイミング等によって像を反転
 - 奇数枚の絞り羽根で必須
 - 絞りを通ってからセンサまでに光束がクロスすると反転
 - 実用上は乱数で絞り形状を点対称に反転

ゴーストサイズの傾向

ゴーストの光源/画面中央からの距離に依存 大雑把な傾向として画面中央寄りの方が小さい



Referenced from 『カメラのやっかいなあいつ。フレア・ゴーストの防止方法。』 Wise Camera <u>https://camera-beginner.sakura.ne.jp/wp/?p=669</u>

ズーミングによる変化

被写界深度ボケの錯乱円径 [Kawase 2010]

- CoC(z):特定のz(距離)における錯乱円径
- CoC(z) = VD / L * (z L) / z

D: 絞り有効径 V: レンズからのフィルム距離 L: 現在のフォーカス距離 z: 計算対象とする距離



- ズーム率をRとすると
 - フォーカスや実効F値が同じならDとVがそれぞれR倍される
 - CoC(z) = R² * VD / L * (z L) / z
 - ズーム率Rの2乗に比例してCoCが変化

ゴーストサイズの変化

- 明確なフォーカスも無くより複雑な変化
 常に大きな収差でボケているような状態
 - ゴースト毎に異なる
 ほとんど変化しないものから大きく変化するものまで
 ズーム率の1~2乗程度で変化させると自然
- もう少しシミュレーションする?
 ゴースト毎に仮想的な焦点距離を想定してCoC計算
 フォーカス時も収差が大きくある程度ボケているとみなす

レンズ構成による変化

- ズーミングやフォーカシング
 - 光学系(レンズ)内の一部のレンズ群が非線形に移動
- ゴーストの位置とサイズにも影響
 - ある程度まとまったグループでゴーストが大きく変化
- 実装
 - ー 画角によって非線形に変化
 ・ゴースト位置を定義するスケーラs
 ・ゴーストのサイズ



ゴーストの輝度に影響する要素

- ARコートの反射率に比例
- 光源の光度に比例
- ゴーストの面積に反比例

ゴーストの輝度に影響する要素

- 光源の光度
 - 光源の輝度に比例
 - 光源に依存
 - 光源の面積に比例
 - ズーミングにより光源のサイズが変化
 - 光源面積はズーム率R²に比例
 - •光源輝度の変化と区別すること(画角では輝度は変化しない)

ゴーストの輝度に影響する要素

- ゴーストの面積に反比例
 - ゴースト毎の基準サイズ(乱数や制御パラメタで決定)
 - 絞りの状態による変化
 - 被写界深度ボケと同様、露出が同じとみなせる場合
 - 絞りで露出が変わるなら最終的なゴースト輝度は変化しない

- ゴースト描画時の露出を適切に考慮する必要あり

- ズーミングによりゴーストのサイズが変化
 - •ゴースト面積はズーム率R2~4程度に比例(前述)

- ゴーストの位置等による変形(後述)

鏡筒によるマスク(ケラレ)

鏡筒によってゴーストがマスクされる

- 被写界深度ボケの口径食(ケラレ)と同じ現象
- 画面端でゴーストが消失するのはケラレが原因
 - イメージサークルに相当
- ゴースト表現では非常に重要な効果
 - 絞りや個々のゴーストの状態によって様々な変化
- 鏡筒のあらゆる位置でケラレが発生する可能性あり
 - 開放絞りでは多くは常にケラレている
 - 正確な表現はシミュレーションでしかほぼ不可能
 - 実用上は正確な表現は不要

絞り同様に距離情報から円形マスクを生成

- 一径食の大きさと動きの関係 [Kawase 2012]
- ゴースト毎にマスクの挙動を決定
 - 動きを制御したいものはマスク半径やケラレ位置で定義
 - 例えば画面中央からどの程度の場所でケラレるか
 - ゴーストの内側からケラレるか外側からケラレるか
 - 殆どのゴーストは乱数で良い
 - 光源が画面端に達した状態である程度ケラレていると実用的

口径食の大きさと動き [Kawase 2012]

- 前ボケの上端が青の点線から下端が赤の点線まで移動すると
 - ケラレマスクの上端が青の点線から上端が赤の点線まで移動
- 後ボケの下端が青の点線から上端が赤の点線まで移動すると
 - ケラレマスクの下端が青の点線から下端が赤の点線まで移動
- かつ、ボケが画面中央にある場合は
 - ボケの中心とケラレマスクの中心は一致(同じく画面中央)



65

ゴーストの口径食の大きさと動き

どちら側から欠けるかによって挙動が異なる



n

ケラレマスクの大きさと位置の関係

P(x,y): 画面中央を原点とするゴースト(処理中のピクセル)の中心座標 r: 欠け始めから完全にマスクされるまでの距離の半分 d: 画面中央から半分欠ける位置までの距離 b: ゴースト半径(外側から欠ける場合は負の値) e: ケラレマスク半径(外側から欠ける場合は負の値)

• ケラレマスク半径は
$$e = \frac{db}{r-b}$$

• ケラレマスクの中心座標は $P_e(x,y) = P(x,y) \frac{r}{r-b}$



ケラレ座標を生成してマスク

- 関係式からケラレマスク用の座標を計算
 - どちらの方向から欠けるかは乱数で良い
 - ゴースト同様に距離情報から円形マスクを生成
 - ゴーストに乗算してマスク
 - 完全にマスクされる場合は描画スキップで高速化







ケラレマスクあり 特に光源が画面外へ出ると 徐々にマスクされて消えてゆく


乱反射や屈折によるコースティクス

- ゴーストに発生する円形のハードエッジ
 - おそらくレンズの端や鏡筒で生じる乱反射?
 - 光源が画面中心付近にあると出やすい
- ケラレマスクをエッジ部分のみ強調することで表現できる



Referenced from 『驚愕のレンズフレア、ゴースト現象』 https://kuwashima.jp/hoshino/flare/flare.html

乱反射や屈折によるコースティクス

- ゴーストに発生する円形のハードエッジ
 - おそらくレンズの端や鏡筒で生じる乱反射? - 光源が画面中心付近にあると出やすい
- ケラレマスクをエッジ部分のみ強調することで表現できる



©Silicon Studio Corp., all rights reserved.











10 ···





Silicon Studio ©Silicon Studio Corp., all rights reserved.

分散と色収差

- ・主に2種類の色収差が知られている
 - 軸上色収差
 - 倍率色収差





軸上色収差

- 被写界深度のボケと同様の色収差
 - ただしより複雑な変化が発生する
 - シミュレーション以外での再現は難しい
- 必ずしも正確な収差は不要
 - 光束マップのようなものから適当にフェッチ
 - ランタイムにプロシージャルで適当に着色

事前計算光束マップを使う場合のサンプル [Kawase 2012, 2015.a]

Spherical lens (with no corrections)



Achromatic doublet lens



Apochromatic (APO) lens



ランタイムプロシージャル着色

- 今回は実行時に簡易的なプロシージャル生成
- 補色関係にある適当な2色を決めて距離情報で補間
 - 絞り形状を定義する距離場の値を使用
 - ハードエッジや中央部分のピークを考慮
 - 実用上は殆ど乱数で良い
 - 合計が白になるように色や強度を調整
 - 最後に2回のAR残存反射色で乗算



ランタイムプロシージャル着色

- 今回は実行時に簡易的なプロシージャル生成
- 補色関係にある適当な2色を決めて距離情報で補間

何か線が見える...

- 絞り形状を定義する距離場の値を使用
- ハードエッジや中央部分のピークを考慮
 - 実用上は殆ど乱数で良い
 - 合計が白になるように色や強度を調整
- 最後に2回のAR残存反射色で乗算

マッハバンド

- •距離情報で色を変えるとマッハバンドが発生する
 - 角から中心方向に向かって不自然な線が知覚される
 - 角張っているほど不自然さが目立つ



マッハバンド [Kawase 2009]

明るさが連続でも階調が変化している場所
 - 傾斜の終端部分の外側に本来存在しない明るさの
 帯があるように見える



シュブルール錯視 マッハバンドと混同されやすい

• 明るさが非連続に変化

- 隣接する色との差が強調されて見える



グラデーション有効時のマッハバンド

• 中心に近いほど距離場を円形に生成することで改善



// Distance Field のコードの最後の部分に追加
d = lerp(d, r0, fCircularity); // 外周を多角形から円形に滑らかに補間
d = lerp(r0, d, d*d); // 内部を円形から多角形に補間,中心ほど円形に近く
// これにより軸上色収差の着色時のマッハバンドを抑制できる

Silicon Studio ©Silicon Studio Corp., all rights reserved.



マッノバンドが生じる





中心に近いほど円形に生成

マッハバンドが改善される







もともと目立たない











Chromatic Aberration of Magnification Chromatic Aberration of Magnification Chromatic Aberration of Magnification



- 複雑な反射/屈折により大きな分散が発生
 - ほとんど目立たないものもある
 - 非常に大きく目立つものもある
 - 一部のゴーストだけ分散させれば良い
- 波長毎にゴーストのパラメタを変化
 - 距離情報やサイズ、変形等を少しずつ変える
 - 面倒ならゴーストをコピーして変化させても良い
 - 一部だけならパフォーマンスへの影響は少ない

ゴーストの色収差適用結果

- 軸上色収差
- 倍率色収差



非線形な変形

Silicon Studio ©Silicon Studio Corp., all rights reserved.

非線形な変形

- 複数の非線形な反射/屈折を経るため
 - 同心円方向の変形
 - 放射方向の変形
- 多くのゴーストは単純な線形の変形で充分
 - 画面中心からの距離の非線形な関数でスケール
 - 中心から離れるほど強く変形
 - 同心円方向/放射方向にそれぞれスケール
 - 個々のゴースト単位では線形(一様)な変形で充分
 - ゴースト毎に変形の性質を変える
- 面積の変化に応じて輝度も変化させる
 - 線形の変形なら単純な逆数スケール



変形あり ゴースト単位では線形

> Í Silicon Studio & ↓ E B I ⊆ З

トポロジーが変化するような複雑な変形

- 10%程度のゴーストは複雑な変形を伴う [Lee et al. 2013]
 - 一部が交差して裏返るような変形
 - コースティクス(集光模様)が発生





『驚愕のレンズフレア、ゴースト現象』 https://kuwashima.jp/hoshino/flare/flare.html

Silicon Studio ©Silicon Studio Corp., all rights reserved.

トポロジーが変化するような複雑な変形

- いくつかの特徴
 - 光源が画面中心から離れるほど強く変形
 - 絞りが開くほど強く変形





トポロジーが変化するような複雑な変形

- 二次元の変形関数を定義
 - 距離場マップの元となる二次元座標を変形
 - 光源が画面中心から離れるほど強く変形
 - 絞りが開くほど強く変形
 - ゴースト毎に異なるパラメタで変形
 - 強く変形するゴーストは一部
 - 面積の変化は近接ピクセルとの勾配等から計算
 - フラグメント単位で縮む場所は明るく、伸びる場所は暗く

元となるテクスチャ座標の非線形変形例 RG: テクスチャ座標, B: 輝度スケール



ゴーストの非線形変形例 変形済み座標からの距離場で生成した形状




変形あり 非線形な変形





変形あり 非線形な変形



光源の位置による変化

• 光源が画面中央から離れる程強く変形

非線形な変形 光源位置比較



非線形な変形 光源位置比較



非線形な変形 光源位置比較



絞り状態による変化

• 開放絞りに近いほど強く変形

非線形な変形 絞り比較: f/2.8

1

0

非線形な変形 絞り比較: f/4 非線形な変形 絞り比較: f/5.6

-

回折フレア

Silicon Studio ©Silicon Studio Corp., all rights reserved.

回折限界とエアリーディスク [Kawase 2010]

- 光(電磁波)の波としての性質
 - 開口絞りを通った光束は回折により一点に集光できない
 - 波動光学的な限界により解像力が制限される
 - •回折による干渉パターン(エアリーパターン)を生じる
 - 中央部の明るい円をエアリーディスクとよぶ



http://en.wikipedia.org/wiki/File:Spherical-aberration-slice.jpg http://en.wikipedia.org/wiki/File:Airy-pattern.svg From Wikipedia, the free encyclopedia

点拡がり関数(Point Spread Function: PSF)

- 光学的にはすべての画素に対する畳み込み
 - 結果的に明るいピクセルのみがフレアとなる
 - ※この関数は無収差円形開口条件

http://en.wikipedia.org/wiki/File:Airy-3d.svg From Wikipedia, the free encyclopedia

フーリエ光学

- 波動光学の一種
 - フラウンホーファー回折
 - フレネル回折

フラウンフォーファー回折

- 開口を通った光が非常に遠くに作る像の近似
 - 開口マスクの離散フーリエ変換(Discrete Fourier Transform: DFT)
 - 開口径の1000倍程度以上離れた像であることが前提
 レンズが存在する場合は焦点像の近似を得られる
- フォーカス以外ではフレネル回折による近似が必要
 - より複雑
 - 分数フーリエ変換(Fractional Fourier Transform: FrFT)
 - ボケの複雑な回折パターンも計算できる

Fraunhofer Diffraction [Kakimoto 2015]

$$I_f\left(\frac{x_f}{\lambda f}, \frac{y_f}{\lambda f}\right) = \left|\frac{A}{\lambda f}\right|^2 |\mathcal{F}[t_o(x_o, y_o)]|^2$$



 $I_f \equiv |U_f|^2$: Wave intensity A: Amplitude of incident light $\mathcal{F}[\cdot]$: Fourier transform operator R: Sufficiently large distance $R \gg 50 \text{m}$ for 5mm^2 aperture size and $\lambda = 500 \text{nm}$

Fraunhofer Approximation in a Lens System

$$I_f\left(\frac{x_f}{\lambda f}, \frac{y_f}{\lambda f}\right) = \left|\frac{A}{\lambda f}\right|^2 |\mathcal{F}[t_o(x_o, y_o)]|^2$$

The diffraction image through a lens system can be denoted using a 2D Fourier transform of the object that causes diffraction. [Goodman 1968]



離散フーリエ変換で 事前に回折フレアテクスチャを作成

- フラウンホーファー回折を前提
 - つまり光源にフォーカスしている状態での近似
- 絞り状態毎に開口画像を用意
 - 絞り羽根枚数
 - 絞り具合
 - 円形度合い
 - 複数の結果を補間して使用
 - •距離場と異なり離散的な画像そのものの補間(クロスフェード)
 - 将来的にはリアルタイムにFFT?
 - 絞り形状生成の距離場でもっと微妙な歪みなどの表現も必要

電磁波の振幅/パワー

- ・電磁波の振幅
 - フーリエ変換結果(複素数)の絶対値
 - 実部と虚部のベクトルの長さ
- 電磁波としてのエネルギー
 - 振幅の2乗

開口画像/電磁波振幅/パワー/パワーx40



電磁波のエネルギーからフレアを生成

- ・DFT結果は単一波長の電磁波
 - ある波長のエネルギーのイメージセンサ上での分布を示す
- ・光源のもつ電磁波スペクトルと感度で積分
 - 結果が回折フレアとなる
 - リアルタイムでの厳密な計算は難しい
 - そもそも光源は多くの場合RGB空間で指定される
 - 光源の波長単位のスペクトルは判らない
- 事前に積分しておく
 - 等エネルギー放射(光源E)におけるXYZ刺激値に変換
 - 描画時の光源は光源Eに対するXYZスケールで近似

等エネルギー放射(光源E)における XYZ刺激値を求める

- CIE 1931 等色関数で可視光波長域を積分
 - 例えば360nm~830nm
 - 光源のスペクトルは全波長に渡って一定(光源E)
 - 回折パワー分布画像を波長に比例させてスケーリング
 - エネルギー合計を等しくするため重みはスケール面積に反比例
 - 必要となる分解数は回折テクスチャ解像度に依存

• 例えば1024x1024なら1nm程度で充分

• XYZ刺激値の二次元テクスチャを作成

上: DFT結果の単波長エネルギー 下: XYZ刺激値(E光源可視波長全域の積分)













回折フレアとしてのマッピング

- イメージセンサ上の座標とフーリエ周波数の対応
 - $(x', y') = (u, v) \cdot \lambda \cdot z_0$ [Hecht 2001] λ : 波長 z_0 : 開口からイメージセンサまでの距離
- マッピングサイズ
 - $\frac{N\lambda z_0}{D} = N\lambda Fe$ N: 絞り画像解像度(DFT元になった開口画像径の実ピクセル数) λ : 波長 D: 有効絞り径 Fe: 実効F値 (= $\frac{z_0}{D}$)



マッピング例

- 512x512 開口画像, f/2
 - マッピングの物理サイズ(mm)
 - $N\lambda Fe = 512(N) \times 0.000360(\lambda) \times 2(Fe) = 0.36864(mm)$
 - λは可視波長域の等色関数積分時のスケールに従う
 - 単波長側の360nmを基準に他波長画像を拡大して積分したなら360nm
 - 長波長側の830nmを基準に他波長画像を縮小して積分したなら830nm
 - 仮に35mmセンサの縦幅(24mm)を基準にすると
 - 0.36864 / 24 = 0.01536(1.536%)
 - 画面縦サイズの1.536%の大きさにマッピング

テクスチャ解像度と開口ピクセル径

- マッピングの物理サイズ(NλFe)
 - テクスチャ自体の解像度とは無関係
 - 開口マスク部分の直径が何ピクセルかだけで決まる
 - センサ上の 0.36864mm の範囲にテクスチャ全体をマッピング
 - テクスチャ全体の解像度が1024でも2048でも0.36864mm

- 開口部分が512ピクセル, f/2 である限り

• 実用上は適当なスケール係数を追加

回折フレアとして描画

- 描画輝度
 - 対象光源の(輝度情報を含む)色をXYZ空間に変換
 - 事前計算回折フレアテクスチャと乗算
 - 対象光源の面積に比例
 - •回折フレアのマッピング面積に反比例
 - 結果をレンダリング色空間へ逆変換
- 実用上は適当なスケール係数を追加
 - -物理的にはフレア全域の合計値が光源の総量に一致

Silicon Studio ©Silicon Studio Corp., all rights reserved.

円形7枚羽根絞り

- sRGB色空間でトーンマップ
- 結果画像のハードエッジをスムース化
 - フーリエ変換時の解像度(周波数)で制限されるため - 円形に減衰させてトリミング(スムースエッジ化)
- イメージセンサ上の縦3mm部分の切り出し
 - 回折は実効F値(Fe)のみで決まる
 - イメージセンサ等には依存しない





フレアが小さく整い過ぎている?

- 実際のレンズには傷や
 汚れや加工痕などがある
 - 不規則な回折が発生
 - 分散による着色や 円形に大きく広がる効果



Silicon Studio ©Silicon Studio Corp., all rights reserved.









DFT結果(単波長のエネルギー)





XYZ刺激値(E光源可視波長全域の積分)
















f/22

.

ノイズ追加回折フレアシミュレーション

マッピングサイズが小さい?

- f/2のマッピングサイズは
 - $N\lambda Fe = N \times 0.000360(\lambda) \times 2(Fe) = N \times 0.00072(mm)$
 - •開口ピクセル数×0.00072mm
 - N = 1024でも 0.73728mm
 - ほとんど広がらない

Silicon Studio ©Silicon Studio Corp., all rights reserved.

 3mm 広げるためには 4000ピクセル以上必要

- 3mm は 35mmセンサの1/8



マッピングサイズを大きくしたい

- 開口マスク画像をできるだけ大きくする必要がある
 - DFTだけを非常に大きなサイズで行えば良い?
 - •今回のサンプルでは開放時は5000ピクセル以上で生成
 - DFT結果を 1024ピクセルに縮小してからXYZ刺激値に積分
 - むしろFrFTで実装すべき?
- 実効F値に比例するため絞り込んだ場合は充分大きい
 - f/2など開放に近い絞りで小さくなることが問題
 - 絞り込んだ状態では逆に拡大され過ぎて解像度が落ちる

マッピングサイズを大きくしたい

- 絞りによって開口マスク解像度を調整
 - 理論上は実効F値に反比例させれば常に同じマッピングサイズ
 - 現実には露出が同じなら絞り込む方がフレアは大きく広がる傾向
 - 実効F値に反比例させると絞り込んだ状態でマッピングサイズが足りない
 - 中間程度のバランスが必要
 - 例えば実効F値の二乗根に反比例
 - マッピング範囲が絞りに応じて異なるためエッジが不自然になる
 2レベル補間結果のエッジ部分を円形に滑らかに減衰させる
- 実用上は適当なスケール係数を追加
 一輝度とサイズの任意の調整が現実的

Silicon Studio ©Silicon Studio Corp., all rights reserved.



その他のサンプル

- 円形5枚羽根絞り
 - f/2(開放)から絞り込んだ状態まで
 - イメージセンサ上の縦3mm部分





検証:回折限界との比較

エアリーディスクの大きさ

- d はエアリーディスクの直径
- λは光(電磁波)の波長
- -nは像側を満たす媒質の相対屈折率(通常1.0)
- ただし正弦条件成立時

$$d = \frac{2 \cdot 1.2196\lambda Fe}{n}$$



http://en.wikipedia.org/wiki/File:Airy-pattern.svg From Wikipedia, the free encyclopedia

Silicon Studio ©Silicon Studio Corp., all rights reserved.

1024x1024の1/10サイズの円形開口画像から FFTで生成したエアリーパターン



生成結果のエアリーディスク径(拡大図) 約24ピクセル強であることが判る



エアリーディスク径の比較

- イメージセンサ上でのマッピングサイズ
 - マッピングの物理サイズは $N\lambda Fe = 1024 \times 0.1\lambda Fe$
 - その中で~24ピクセル径の占めるセンサ上での物理サイズ
 - $\sim 24/1024 \times 1024 \times 0.1\lambda Fe = \sim 2.4\lambda Fe$
- 回折限界からの計算
 - エアリーディスク径は
 - $2 \times 1.2196 \lambda Fe = 2.4392 \lambda Fe$
- ほぼ理論どおりに一致している

生成結果サンプル

Silicon Studio ©Silicon Studio Corp., all rights reserved.

イメージベースとスプライトベース

- いずれも1/2x1/2縮小バッファ上で生成
- 回折フレアは35mmセンサ(縦24mm)の4倍で計算
 つまり縦6mmのイメージセンサに相当

イメージベース 全てのピクセルから発生



スプライトベース 特定の光源のみ



ハイブリッド



円形5枚絞り

- 絞り状態の比較
 - 開放から絞り込んだ状態まで
 - 絞りの形状の変化
 - 非線形な変形
 - 回折フレアの大きさと形状

- 露出は一定のためゴースト輝度はゴースト面積に反比例

• 1080pフル解像度で生成

5枚絞り f/2.0 (開放)

















.





C. C. C. C. C. C.

•





Ch is the set

.





· · · · · ·





....



円形5枚絞り

• 絞りに応じて露出を変えた場合

5枚絞り f/2.0 (開放)



5枚絞り f/2.8



5枚絞り f/4














その他実装に関するノウハウ

その他実装に関するノウハウ

- レンズフレアのカスタマイズ
- ・物理的整合性の度合いの調整
- ・ゴーストLOD

レンズフレアのカスタマイズ

- レンズフレアの全体的な制御
 - どの程度のサイズか
 - 位置の偏り方
 - ばらつき方
 - ARコートの反射特性
 - 反射スペクトルを何種類か用意
 - それぞれの使用頻度を制御パラメタに
 - 例えば殆ど青系で稀に緑系のコートがある等
 - 色収差の程度

レンズフレアのカスタマイズ

- 乱数のシードでバリエーションを生成できる
 多くのゴーストを乱数ベースで生成するため
- クォリティと負荷の制御
 - 乱数ベースのゴースト数や分散の波長分割数

主なゴーストをカスタム定義 残りを自動生成(1)



主なゴーストをカスタム定義 残りを自動生成(2)



グローバル設定と 乱数による自動生成のみ(1)



グローバル設定と 乱数による自動生成のみ(2)



主なゴーストをカスタム定義 絞りを特定の値に固定 ノンフォト調レンズフレア



物理的整合性の度合いの調整

ズーミングによる変化

ゲーム用途では極端に大きなズーミングがあり得る
 ・大口径で100倍ズーム等
 物理的整合性によって問題が発生するケース
 ・超ズームアップによりゴーストが薄く大きくなり過ぎる
 - ほとんどのゴーストが事実上消えてしまう

ズーミングによる変化

- - ゴースト計算用の画角をカメラ設定とは別に設計する
 - 変化の度合いを指定できるようにすると実用的
 - 例えばズーム率Rのn乗を計算上のズーム率として適用
 - n = 1.0 なら物理的に追従
 - n = 0.5 ならある程度追従
 - ズーム率を0.5乗すると100倍ズームが10倍相当となる
 - n = 0.0 ならゴースト用の画角が一切変化しない
 - 物理ベースのレンズフレアでも同様の応用可能

ゴーストしの

- 画角変化を抑える方法では対処できない場合
 あまりにもズーム率が大きすぎる
 - 画角変化への追従が少なすぎると不自然になるケース
- ゴーストのLOD化
 - ある程度大きく薄くなったゴーストをフェードアウト
 位置と大きさを小さくしたゴーストをフェードイン
 - 描画する枚数は同じで画質を向上できる
 - •非常に大きくて薄い無駄な描画を削減できる

LOD Off / On 広角時 : 変化なし









LOD Off 大きくズーム時

፤ Silicon Studio &♀♀ ㅌ ㅌ ι ⊆ З

LOD On 大きくズーム時

Ĩ Silicon Studio & ↓ E B I 5 3

100



まとめ

- スプライトベースとイメージベース
- 近年の物理ベースレンズフレア
 - レンズデータが必要
 - 負荷とクォリティのバランスが難しい
- レガシーなスプライト手法に光学効果を考慮
 - 自由なカスタマイズ可能
 - ほとんどのパラメタは乱数で自動生成
 - 絞りやズームによる光学的な挙動を考慮
 - 回折フレアを追加
 - 物理的整合性は必要に応じて変化



- 可能な範囲でより正確な光学表現
 - ゴーストの位置やサイズ、並び方
 - フォーカシング、ズーミング時の挙動
 - FrFTを用いた回折リングの表現
 - アナモルフィックレンズ
 - これは簡単に応用可能
- イメージベースのスキャッターへ
 - 反射光や面積をもつ光源への対応
 - モーションブラー等も適用可能
- 回折フレア
 - 実行時にFFTで生成
 - 事前計算不要
 - フレアテクスチャの切り替えが不要
 - 光源の位置によって変化するスターバースト表現



ご質問は?

masa@siliconstudio.co.jp



References

References

[Goodman 1968]	Goodman, J. W. "Introduction to Fourier Optics." McGraw-Hill ,1968.
[Hecht 2011]	Hecht, E. "Optics." Addison Wesley, 2001.
[Hullin et al. 2011]	Hullin, M., Eisemann, E., Seidel, H., Lee, S. "Physically-Based Real-Time Lens Flare Rendering." <u>http://resources.mpi-inf.mpg.de/lensflareRendering/</u> SIGGRAPH 2011, 2011.
[Kakimoto 2015]	"Wave optics based glare generation techniques." Real-time Rendering of Physically Based Optical Effect in Theory and Practice. SIGGAPH 2015 Course, 2015.
[Kawase 2003]	Kawase, M. "Frame Buffer Postprocessing Effects in DOUBLE-S.T.E.A.L (Wreckless)." GDC 2003, 2003.
[Kawase 2009]	Kawase, M. 『アンチ・縮小バッファアーティファクト』 <i>CEDEC 2009,</i> 2009.
[Kawase 2010]	Kawase, M. 『魅力ある絵作りのために知っておきたい色光学豆知識』 <i>CEDEC 2010,</i> 2010.
[Kawase 2012]	Kawase, M. 『実践!シネマティックレンズエフェクト』 CEDEC 2012, 2012.
[Kawase 2015.a]	Kawase, M. "Making Your Bokeh Fascinating." Real-time Rendering of Physically Based Optical Effect in Theory and Practice. <i>SIGGAPH 2015 Course</i> , 2015.
[Kawase 2015.b]	Kawase, M. "Subtle Anamorphic Lens Effects." Real-time Rendering of Physically Based Optical Effect in Theory and Practice. SIGGAPH 2015 Course, 2015.
[Lee et al. 2013]	Lee, S., Eisemann, E. "Practical Real-Time Lens-Flare Rendering." EGSR 2013, 2013.
[Mittring 2012]	Mittring, M. "The Technology Behind the 'Unreal Engine 4 Elemental demo'." Advances in Real-Time Rendering in 3D Graphics and Games. SIGGAPH 2012 Course, 2012.

Special Thanks

• シリコンスタジオ株式会社 ミドルウェア技術部 疋田 祥太郎