

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2004-511863
(P2004-511863A)

(43) 公表日 平成16年4月15日(2004.4.15)

(51) Int. Cl. ⁷	F I	テーマコード (参考)
G06T 1/00	G06T 1/00 500A	5B050
G06T 11/80	G06T 11/80 A	5B057

審査請求 未請求 予備審査請求 有 (全 74 頁)

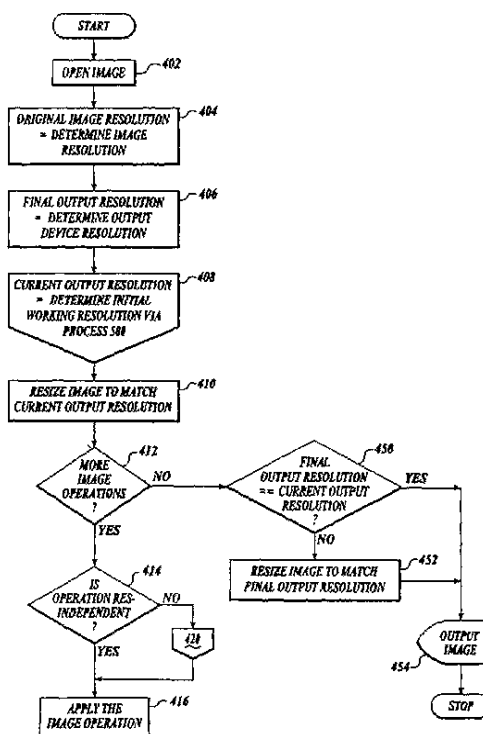
(21) 出願番号	特願2002-535064 (P2002-535064)	(71) 出願人	503138008 ピクチャーアイキュー インコーポレイテッド アメリカ合衆国 98104 ワシントン州 シアトル フィフス アベニュー サウス 505 スイート 350 アイビー カウンセル
(86) (22) 出願日	平成13年10月12日 (2001.10.12)	(74) 代理人	100079049 弁理士 中島 淳
(85) 翻訳文提出日	平成15年4月14日 (2003.4.14)	(74) 代理人	100084995 弁理士 加藤 和詳
(86) 国際出願番号	PCT/US2001/042694	(74) 代理人	100085279 弁理士 西元 勝一
(87) 国際公開番号	W02002/031754		
(87) 国際公開日	平成14年4月18日 (2002.4.18)		
(31) 優先権主張番号	60/240,495		
(32) 優先日	平成12年10月13日 (2000.10.13)		
(33) 優先権主張国	米国 (US)		
(81) 指定国	EP (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), JP, US		

最終頁に続く

(54) 【発明の名称】 画像の解像度非依存型レンダリングのための手法

(57) 【要約】

通常は解像度依存の画像動作と考えられる画像動作を実質的に解像度非依存の動作に変換することによって、画像データ(402)レンダリングの性能を向上させるための方法。これは、レンダリング(454)が、例えば、画面上の表示に対し使用される低解像度画像データ上で実行できることから、著しい性能の向上を可能とし、画像効果がより高い解像度のレンダリングに適用される場合、その効果は視覚上、より低い解像度で見られる効果と略同一である。このような通常は解像度依存の画像動作を擬似解像度非依存型動作に変換することは、画像がより高い解像度でレンダリングされる時、適用される画像効果が、より低い解像度において効果が有する場合と同じ外観を実質的に有するという確信とともに、適用される画像効果をより低い解像度の画像上で実行させることもできる。



【特許請求の範囲】**【請求項 1】**

画像効果の外観を実質的に維持しながら、結果として生じるいかなる解像度においてもデジタル画像のレンダリングを可能とするように、普通は解像度に依存する画像効果をデジタル画像に適用するための方法であって、

(a) 異なる解像度でレンダリングされる時にどの画像効果パラメータが画像効果の結果を実質的に修正するかを決定するステップと、

(b) 結果のレンダリングを得るために前記パラメータを修正するステップと、を含む、方法。

【請求項 2】

前記画像効果が前記画像のヒストグラムから導出される変換関数に基づくとともに、

(a) 元の解像度における前記画像の画素データを分析するステップと、

(b) 適切な画像操作パラメータを決定するステップと、

(c) 所定の結果の解像度に従って前記パラメータを修正するステップと、

をさらに含む、請求項 1 に記載の方法。

10

【請求項 3】

前記画像効果がデジタル画像の特定の領域の座標及び半径に基づくとともに、

(a) 元の解像度に関連して座標データ及び半径データを保存するステップと、

(b) 結果の解像度によって修正される半径情報及び座標情報とともに、該結果の解像度で所望の画像効果を実行するステップと、

をさらに含む、請求項 1 に記載の方法。

20

【請求項 4】

前記画像効果が赤目軽減である、請求項 3 に記載の方法。

【請求項 5】

前記画像効果が人工照明の調整であり、この人工照明の調整が、

(a) 画像における光周波数の各範囲の量のスペクトル解析と、その量の割合を決定することと、

(b) 前記割合を用いて画像効果を適用することと、

(c) いかなる解像度で画像をレンダリングする時も、前記決定された光の割合が略一定に維持されることと、

を含む、請求項 1 に記載の方法。

30

【請求項 6】

前記画像効果が、画像解像度を決定することに依存する値に基づくとともに、要求される前記結果の解像度に基づいて該依存値をスケールリングするステップをさらに含む、請求項 1 に記載の方法。

【請求項 7】

前記画像効果が、異なる不連続の内部パラメータ値に基づき、これによって、要求される前記結果の解像度に依存する適切な値の部分集合の選択を要求する、請求項 1 に記載の方法。

【請求項 8】

(a) 画像を開くステップと、

(b) 画像解像度を決定するステップと、

(c) 出力装置解像度を決定するステップと、

(d) 画像効果を適用する最適な解像度を選択するステップと、

(e) 選択された前記最適な解像度に対し前記画像効果を適用するステップと、を含む、デジタル画像の解像度依存特徴に対し手法を適用する方法。

40

【発明の詳細な説明】**【0001】****【発明の属する技術分野】**

本出願は、2000年10月13日出願の米国仮出願シリアル番号第60/240,495

50

5号の利益に対する権利が付与される。

【0002】

本発明は概してデジタル画像処理システムに関する。詳細には、本発明は画像データをレンダリングする性能を大きく向上させる手法を説明する。さらに詳しくは、もともと解像度非依存であると見なされていない操作に対しても解像度非依存型特徴を利用する手法が説明される。さらに具体的に言うと、画像操作がいずれの解像度に適用される時も、解像度非依存型動作をシミュレートする（擬似解像度非依存型と称する）ために一定の手法が適用される。

【0003】

これら手法は、パーソナルコンピュータ、デジタル画像民生用装置上で実行するようなコンピュータアプリケーションにおいて、そして（物理的に配線された又は無線の媒体上のクライアント側又はサーバ側の実行可能なコードの形式で）ネットワーク環境内で写真を表示し操作する時に、レンダリングシステムの性能を大きく向上させることができる。

10

【0004】

ほとんどの画像操作は2つのカテゴリ、即ち、画像の解像度に関係なく同じ結果を生成するもの（即ち、解像度非依存式）、及び画像データの異なる解像度に対し著しく異なった目に見える結果を生成するもの（即ち、解像度依存式）とに分類される。例えば、回転はすべての解像度にわたって不変であり、解像度に依存していない。しかしながら、フレスコ画法効果等の特殊効果の画像処理操作は、異なる解像度にわたって広範囲に変化し、解像度に依存している。

20

【0005】

解像度非依存の操作のみがサポートされると、出力のスクリーン上の表示に必要とされる低解像度画像データ上でレンダリングが実行可能であるので、著しい性能の向上が達成される。なぜなら、一般にディスプレイの解像度は高解像度プリンタ等の他のほとんどの出力装置より通常ははるかに低いからである。

【0006】

解像度に依存する画像操作のすべて又は少なくとも大多数が解像度非依存型動作をシミュレートするような方法で動作すると、解像度非依存型の利点のほとんどが依然として実現される。解像度非依存の方法で解像度の連続範囲をサポートすることが不可能なことがある一方、解像度非依存型に近似させる一定の仮定及び制約が行なわれ、これによって新しい種類の擬似解像度非依存型画像操作を生成する。好ましい実施の形態では、すべての操作が解像度非依存型であるか、又は擬似解像度非依存型である。

30

【0007】

本発明の目的の1つは、低解像度の画像データを効果的に処理できることであり、すべての解像度にわたって一貫した結果を得られることである。これが表示画面装置において画像を表示し処理する時に特に重要であるのは、一般にこれら装置が画像データの解像度より、はるかに低い解像度であるからである。さらに、本発明は、たいていのアプリケーションが今日では正確にサポートすることが出来ない、すべての解像度における一貫した結果を犠牲にすることなく、このような効率性を提供する。

【0008】

本発明の別の目的は、ネットワーク接続環境において極めて重要な、その他の場合に必要となるよりはるかに少ない画像データを処理し送信できることである。クライアントが、局所的に（クライアント側で）処理される画像データをダウンロードしなければならない場合、少量のデータをダウンロードするだけが要求されることは、非常に有益である。今日では、小さい画面の解像度表示（元の解像度の画像より小さい）が必要であっても、画像操作が一貫した結果を得られるように、元の解像度の画像は局所クライアントに送信されなければならない。これら操作がサーバ上で実行される時でも、画像データの画面上の解像度で画像操作を実行する能力がサーバの処理及びメモリ要件を著しく軽減し、これにより、同じ時間内で更に多くのトランザクションを実行させる。

40

【0009】

50

これら手法により、処理能力が低くメモリがほとんどない低コスト家庭用電化製品の画像装置上でデジタル画像化操作を実行できることは、同様に重要な利点である。今までのところ、これはシステム制約により実現が困難であった。

【0010】

【従来の技術及び発明が解決しようとする課題】

アドビフォトショップ（商標）等の従来の画像処理アプリケーションは、すべての画像処理操作に対し一貫した結果を得るための最良の方法が、目標とされた出力装置の解像度（元の画像解像度よりも高いか低いか）については関係なく元の画像解像度で操作を実行することであると仮定する。これにより、画像が低解像度ビデオディスプレイ又は高解像度プリンタで表示されるかに関係なく、一貫した結果が保証される。従来の画像処理アプリケーションは通常、以下に示す2つのアプローチをサポートする。

10

【0011】

図1Aに示される第1のアプローチは、更新された出力画像が出力装置から要求される場合、適用される画像操作のシーケンスを蓄積し、各操作を再びレンダリングする。これは、画像の元の解像度におけるすべての画像操作の完全な再処理を伴う。これは、各操作が元の画像解像度に再適用されるので、非常に時間のかかる処理である可能性がある。これは、各操作が操作のリストに記憶されるので、無制限の取消し機能を有する柔軟性をユーザに付与する。

【0012】

図1Aを参照すると、プロセス100は、従来の画像編集モデルを用いたデジタル画像の処理を詳述するフローチャートである。プロセス100は102から開始し、特定のデジタル画像が該画像の解像度においてアプリケーションによって開かれる。再サンプリング/再スケーリングは発生しない。104では、画像編集操作が実行されるべきか否かが決定される。実行すべき画像操作がない場合、制御は108に渡される。そうでない場合、106において、画像操作が元の画像解像度データ上で実行され、制御は104に戻され、処理すべき更なる画像操作があるか否かが判断される。このモデルにおいて、連続する画像操作が画像に適用された前の画像操作の上に適用（又は蓄積）されることは留意すべきである。

20

【0013】

108において、所望の出力解像度が決定される。110では、元の画像解像度がその所望の出力解像度に一致しない場合、112において画像は一時バッファに複写され、所望の出力解像度に一致するようにサイズ変更され、さらに制御は114に渡される。そうでない場合、制御は直接114に渡され、画像は出力装置に送信される。選択された出力装置がディスプレイである場合、出力解像度は画像がディスプレイ上で拡大又は縮小（ズームイン/ズームアウト）されるかによって変化することがある。しかしながら、注目すべきは、画像操作が概して元の画像解像度データに適用され、すべての操作が適用された後に、画像がその出力装置の解像度に一致するようにサイズ変更されることである。

30

【0014】

図1Bに図示される第2のアプローチは、元の画像解像度の画像データ上で、但し、中間作業バッファ又はキャッシュしたバッファにおいて各操作を実行する。この作業バッファは、元の画像解像度の画像全体を含む。新しい操作がそれぞれ適用されると、作業バッファ上で直接実行される。所望の出力解像度画像が生成される時、作業バッファは所望の出力解像度に一致するように再サンプリングされる。明らかに、各操作が通常は作業バッファにおいて一度適用され、新しい出力解像度が要求される時に再処理される必要がないことが利点である。このアプローチでの欠点は、無制限の取消し/やり直し操作が可能である一方、更新された出力画像が要求される時に画像操作のリスト全体が通常は再レンダリングされないため、こうした特徴をサポートすることはさらに困難であるということである。

40

【0015】

図1Bを参照すると、プロセス150は、このモデルを用いたデジタル画像の処理を詳

50

述するフローチャートである。プロセス150は、152から開始し、特定のデジタル画像がアプリケーションによって開かれる。154では、デジタル画像が元の解像度と同じ解像度で作業バッファに複写される。156では、画像編集操作を実行すべきかが決定される。実行すべき画像操作がない場合、制御は160へ渡される。そうでない場合、158において、画像操作が(蓄積された操作とともに)作業バッファ上で実行され、制御は156へ渡され、処理すべき更なる画像操作があるか否かが判断される。

【0016】

160において、所望の出力解像度が決定される。162では、(元の画像の解像度である)作業バッファの解像度が所望の出力解像度に一致しないと、164において、作業画像が一時バッファに複写され、所望の出力解像度に一致するようにサイズ変更される。い
10
ずれの場合も、166において一時バッファからのサイズ変更された画像が出力装置上に表示される。プロセスは、画像を再レンダリングする要求又はユーザによる新しい画像操作の要求が適用されるまで168において待機する。170では、プログラムが終了すべきかを判断し、そうであれば、適切なステップが採用され、プロセスが中断する。そうでない場合、要求を処理するために制御は156へ戻される。このように、画像操作のリスト全体は作業画像バッファに再び適用される必要はない。

【0017】

アドビフォトショップ(商標)によって採用されるアプローチは2つのアプローチの組み合わせである。この場合、中間作業バッファが使用されるが、これは、ユーザが取消し操作を実行するとフラッシュ可能なキャッシュと考えられる。作業バッファは廃棄可能であ
20
るとともに、画像操作の更新されたリストに基づいて再生成可能である。あるいはまた、作業バッファは作業バッファの以前にキャッシュされた複写に戻されることができる。極端なことを言うと、すべての中間操作がキャッシュ可能であり、無限の取消し/やり直し操作の効果をシミュレートする。これは、かなりの量のメモリ及び/又はディスクスペースを使用するという犠牲を払う。

【0018】

これらアプローチのそれぞれが、すべての解像度にわたって一貫した結果を付与する利点を有すると同時に、画像処理操作が元の画像解像度のすべての画素上で実行されなければ
30
ならないので、概して処理が遅くなる。中間作業バッファキャッシュの使用により、新しいレンダリング解像度が要求されるたびにすべての画像操作を再処理する必要性を最小限にしているが、画像操作は依然として最初は元の解像度で実行されなければならない。今日の最速プロセッサにおいても、これは依然として時間のかかる操作である。

【0019】

回転又は色調整フィルタのような解像度に依存しない操作が実行される時に、このモデルの変形が可能であることは、留意すべきことである。このような状況において、作業バッファを、操作のプレビューが示される画面解像度のような、はるかに低い解像度にサイズ変更/再サンプリングすることは一般的である。この手法を使用すると、限られた解像度非依存型操作の集合に対しリアルタイム操作が実現可能である。ユーザがその結果に満足
40
すると、操作は元の解像度の作業画像バッファに適用される。

【0020】

これにより上述された不利点の一部が緩和されると同時に、画像操作の小さな部分集合のみがこの解像度非依存型動作に一致し、あるいは所望のリアルタイム操作の機能性をサポート
40
することができる。

【0021】

もっと最近では、すべての画像操作が解像度に依存しない方法で実行される時に有効な画像レンダリングを可能とする新しい手法が利用可能である。デジタル・イメージング・グループ(DIG)を構成する企業の共同体によって開発されたフラッシュピクス(FlashPix(商標))手法をサポートする画像アプリケーションは、画面上の画像を一貫した方法で低解像度で表示し操作することができるが、但し、高解像度画像データを処理する能力を保持する。
50

【0022】

フラッシュピクス手法は、回転、変換、クロッピング、カラーツイスト、ぼかし/シャープネス調整、及び明るさ/コントラスト調整を含む画像操作の限られた組を定義する。これらの操作はそれぞれ、フラッシュピクス(商標)規格によって定義されるような特定の「2の累乗」解像度で実行可能であるように定義されるが、このようなすべての解像度にわたって一貫した結果を付与する。

【0023】

図2を参照すると、プロセス200は、フラッシュピクス(商標)画像モデルを用いたデジタル画像の処理を詳述するフローチャートである。プロセス200は202から開始し、特定のデジタル画像がアプリケーションによって開かれる。204において、所望の出力解像度が決定される。205において、出力解像度と元の画像解像度が同じかどうか判断される。それらが同じでない場合、206において、画像は出力装置の解像度に一致するようにサイズ変更される。いずれの場合にも、208では、画像編集操作が実行すべきかどうか判断される。実行すべき画像操作がない場合、制御は212に渡される。そうでない場合、210において、画像操作が206でサイズ変更された画像データ上で実行される。制御は208に戻され、実行すべき異なる画像操作があるかどうか判断される。

10

【0024】

212において、処理された画像は出力表示装置に送信される。204において出力解像度が既に決定され、すべての処理が目標とされた出力解像度上で実行されたので、画像の異なる再サンプリングは必要ではない。上記で述べたように、同様の作業バッファキャッシュがフラッシュピクス(商標)レンダリングモデルにおいても使用されることは留意すべきである。

20

【0025】

フラッシュピクス(商標)画像モデルは、低解像度画像データ上で実行される時に画像操作の迅速且つ有効な処理という利点を付与する。これは低解像度出力表示装置上のリアルタイム操作と、リアルタイムレンダリングを提供する。これは明らかに有利である一方、解像度に依存しない操作の小さな集合のみが実際にサポートされる。より詳細には、アーキテクチャが本明細書に述べられたような条件を許可しないので、解像度非依存型、又は擬似解像度非依存型の操作をサポートすることは不可能なことがある。フラッシュピクス(商標)が特定の「2の累乗」解像度をサポートするのみであり、本発明によって提供されるような連続範囲の解像度をサポートしないこともまた留意すべきである。

30

【0026】

画像操作の解像度要件が無視され、各操作がある一定の解像度で処理される正反対の単純化したアプローチもさらに使用される。この解像度は通常、元の画像解像度に一致せず、また操作が異なる解像度に適用された場合に一貫した動作を提供するものでもない。画像操作が解像度非依存である場合に対し、フラッシュピクス(商標)によってサポートされるようなものに関しては、このモデルが作動し、一貫した結果を付与する。しかしながら、画像操作が解像度非依存でない場合に対し、さらに一貫性のない結果が見られる。非一貫性の程度は特定の画像操作の関数である。

40

【0027】

例えば、ローパス5x5カーネル等の解像度依存型ぼかし操作を適用する場合、ぼかしの量は解像度(及び画像サイズ)が高くなるほど目立たなくなる。これは、いずれの場合も画像がある程度不鮮明な状態に見えるのでそれほど不快感をもたらすことはない。しかしながら、湖水に落ちる石の波紋のシミュレーションが画像にレンダリングされる「リップル」効果に対して、同心性の波紋の実際の数と外観は、解像度、及びアルゴリズムの正確な特徴によって変化する(後述されるリップル動作のサンプル出力を参照)。

【0028】

図3を参照すると、プロセス300はこのアプローチを用いたデジタル画像の処理を詳述するフローチャートである。プロセス300は、302から開始し、特定のデジタル

50

画像がアプリケーションによって開かれる。304において、所望の出力解像度が決定される。305では、出力解像度と元の画像解像度が同じであるかどうか判断される。それらが同じでない場合、306において画像が出力装置の解像度に一致するようにサイズ変更される。どちらの場合にも、308では、画像編集操作を実行すべきかどうか判断される。実行すべき画像操作がない場合、制御は312へ渡される。そうでない場合、310において画像操作がステップ306でサイズ変更された画像データ上で実行される。これは、たとえ画像操作が元の画像解像度データを要求する場合でも発生し、このため、適切な結果が得られないことがある。制御はその後、308に戻され、更なる画像操作を実行すべきかどうか判断される。

【0029】

10

312において、処理された画像が出力表示装置に送信される。304では出力解像度が既に決定され、すべての処理が目標とされた出力解像度上で実行されたので、画像の更なる再サンプリングは必要ではない。上述されたように、同様の作業バッファキャッシュがこのレンダリングモデルにおいても使用可能であることは留意すべきである。

【0030】

この種のアプローチが画像処理操作をサポートするウェブサイト上で採用されることは一般的であるが、主要な出力目標は低解像度出力表示装置である。一般に、画像操作が（低解像度の）画像に適用された後で、その結果は通常、電子メールで受信者に送信される。受信者は、低解像度出力表示装置上で（既にラスライズされた）低解像度結果を見る。

【0031】

20

例えば、オンライン印刷フルフィルメント・サービスを介して操作が印刷用の元の解像度画像データに再適用される場合には、その状況はさらに多くの問題がある。ユーザは低解像度でレンダリングされた画像の画面を介して出力をプレビューすることもできる。この場合、ディスプレイに表示された出力は注文された最終印刷出力に一致しないことがある。

【0032】

80%までこの矛盾が検出されなくても、その違いに気付く者にしては、印刷出力の払い戻しを要求することがある。この状況は非常にコストがかかり、明らかに満足できる解決ではない。本発明はすべての操作に対しすべての解像度にわたって一貫性が実現されるので、より望ましいの結果を生成し、これにより顧客満足度を向上させる。

30

【0033】

さらに重要なことには、「イグノア・イット」アプローチを採用するフォトサービスがこうした問題をできるだけ少なくするためにサポートされた画像操作をたいがい制限するだろうということである。これら画像操作が解像度非依存の、又は解像度非依存の特徴を利用可能であるように構成される（即ち、擬似解像度非依存型）かを仮定すると、より広範囲の画像操作を提供できることが望ましい。

【0034】**【課題を解決するための手段】**

要約すると、本発明は、画像の最終解像度に関係なく、効果が略同一の外観を有するように標準的に解像度依存の画像効果を適用するための方法から構成される。これにより、レンダリングされた後の解像度に関係なく、結果として得られる効果が元の解像度に適用された時と同じ外観を有するであろうという確信をもって画像効果が画像解像度に適用されることができる。どの特定の効果パラメータが解像度に依存するかを決定し、レンダリングされている最終画像解像度上で予測される修正値でパラメータを修正することによって、解像度依存パラメータを実質的に解像度非依存型パラメータに変換することができる。

40

【0035】**【発明の実施の形態】**

本発明は、その更なる利点とともに、添付図面に関連して解釈される以下の説明を参照して最も良く理解することができる。

50

【0036】

本発明は、一度に数百から数千の同時操作を処理するシステム上に配備されるASP（アプリケーションサービスプロバイダ）モデルの一部として、画像処理サーバ上の配置をもとも対象としている。多くのコンピュータにわたって拡張可能な解決が望ましい。より重要なことには、レンダリングされた画像のサイズに関係なくすべての画像の解像度にわたって一貫した結果を提供するが、出力解像度が低下するにつれて、レンダリングする時間が減少する。

【0037】

画像データが低解像度（例えば、320×240）画像及び数メガ画素画像（例えば、3000×3000画素）の範囲内で変動することは留意すべきである。このような非常に大きな画像のレンダリングを可能にするモデルが開発され、但し、低解像度画像データ（例えば、320×240）上で実行されると、著しい性能の向上が実現される。これにより、画像が迅速に表示されるのみならず、（画像データが存在する）データ・ストレージ・サーバ、（画像処理コードが操作をレンダリングする）画像処理サーバ、及び（出力が見られる）クライアントシステムの間で送信されなければならない画像データの量が低減される。

10

【0038】

これにより、幾つかの利点が付与される。まず、ごく少量の画像データが実際に処理されるので、各操作ごとの処理能力の量が大幅に低減される。さらに重要なことには、ネットワーク環境では、低解像度画像データのみがサーバと遠隔クライアントコンピュータとの間で送信されなければならない。これは、ローバンド幅（即ち、56Kbモデム）環境で実行する時に特に重要である。

20

【0039】

本発明によって実現される別の利点は、解像度非依存、あるいは擬似解像度非依存のいずれかの画像操作を処理する時、元の画像解像度はアクセスされる必要はなく、またさらに利用可能である必要もない。ある状況において、元の画像解像度は利用できず、高解像度の印刷出力の生成が要求される時に利用できさえすればよい。この場合、処理のほとんどが低解像度で、サーバコンピュータや、クライアント/ホストコンピュータのいずれかで起こらなければならない。本発明は、元の画像解像度が利用できなくても一貫した出力が結果として得られる解法を提供する。これがなければ、解像度に依存していない画像操作以外に対して一貫した動作を保証することは不可能であろう。

30

【0040】

低解像度画像の処理は、処理能力の低下とメモリ要件の低減を必要とするという結果になり、これらの必要性は、低コスト家庭電化製品デジタル画像装置を開発する時に重要である。こうしたデジタル画像装置は、情報機器、デジタルカメラ、デジタルカムコーダ、デジタルテレビ、デジタルフォトスキャナ、フォト・イネーブルドな（photo-enabled）セットトップボックス、フォト・イネーブルドなゲーム機、フォト・イネーブルドなインターネットデバイス、セルフォン、ケーブルセットトップ・ボックス、ウェブテレビ（商標）、又は画像を見ることのできるその他のコンピュータ装置を含む。

40

【0041】

解像度非依存型のカテゴリに分類される画像操作は、すべての解像度にわたって一貫した結果をもたらすものを含む。例えば、画像Aでは、画像操作が特定の解像度の画像に適用され、その画像はより低い解像度にサイズ変更される。画像Bでは、画像はまず、より低い解像度にサイズ変更され、その後画像操作が適用される。画像A及び画像Bが、再サンプリング/サイズ変更操作の間に持ち込まれるエラーを考慮し、視覚上十分に近い場合、その操作は解像度に依存していないと考えられる。言い換えると、ユーザが画像Aと画像Bを並列して見る場合、それらの画像は視覚上同じように見えるはずである。

【0042】

このカテゴリーに分類される以下の画像操作の一部は、回転、クロッピング、変換、カラ

50

ー調整、カラーツイスト、明るさ/コントラスト調整（画像のヒストグラムとは異なる）、（フラッシュピクス（商標）画像モデルによって定義されるような）ぼかし/シャープネス操作、デュオトーン（画像を2つのトーンカラー画像に変換する）、グレースケール/黑白、ネガ、露光過度、ポストリゼーション（色の数の減少）、及び二層構造/しきい値を含む。これは決して総合的リストではないが、解像度に依存していないと考えられるタイプの操作を例示する。ここで留意すべきは、フラッシュピクス（商標）画像モデルは、このような操作の小さな部分集合をサポートするだけだということである。

【0043】

各種の擬似解像度非依存の画像処理操作がある。解像度非依存の動作をシミュレートできる方法を決定するために更なる解像度依存の画像操作が調査分析されると、新しい種類の擬似解像度非依存の操作が生成されることになる。

10

【0044】

ヒストグラムベースの画像操作

このクラスの画像操作は、画像のヒストグラムからの変換関数に基づいて所与の画素値を修正する。例えば、自動拡張機能はヒストグラムを見て、ヒストグラムの各端部から5%を切断し、ヒストグラムを範囲全体にわたって再分散する。元の解像度画像のヒストグラムが低解像度画像のヒストグラムに近いので、これは解像度非依存であると考えられる可能性がある。しかしながら、これは一般に正しいが、解像度が非常に低くなる（例えば、 192×192 のサムネイル表示）とその仮定は崩壊する。したがって、画像がレンダリングされるたびに元の解像度画素データの処理を必要とすることなく真の解像度非依存型を提供するアプローチが望ましいとされる。

20

【0045】

本発明は、多重段階において画像操作を実行することによって前記問題を解決する。分析（アナライズ）段階はまず、画像データ上で実行される。この段階は、ヒストグラムが演算可能であるように、理想的には元の画像解像度ですべての画素データを処理する。この段階の間には、画素は修正されない。ヒストグラムが収集されると、ヒストグラムをどのように再分散すべきかというような適切な画像操作パラメータが確認可能である。第2の段階、即ち、フィルタ段階の間に、画像データが実際に修正される。このアプローチの利点は、パラメータが解像度に依存していないので、これにより、フィルタ段階が画像データの元の解像度よりも低いか高いかのいずれかの解像度で実際に実行されることができ

30

【0046】

元の画像データの解像度でアナライズ段階を実行し、今後のレンダリングの結果を記憶させることによって、フィルタ段階はいずれかの解像度で実行可能とされる。この手法は、すべての解像度にわたって解像度非依存の動作をシミュレートする。

【0047】

一般分析段階/フィルタ段階の利用

上述した画像操作のクラスは、このカテゴリーの部分集合である。分析/フィルタ段階モデルがヒストグラムフィルタ操作の使用法に関して上述されている一方で、より一般的な方法で使用可能である。例えば、赤目軽減及び人工照明調整は、普通は解像度に依存すると考えられる2つの画像操作であるが、分析段階を最初に元の画像解像度で実行させることによって解像度非依存型をシミュレートすることができる。

40

【0048】

赤目軽減フィルタ操作は2つの部分、即ち、（a）赤目の検出と、（b）影響を受けた領域の削除/除去、に分割できる。まず第1に、赤目を正確に検出するために、元の画像解像度で検出を実行することが望ましい。そうでない場合、副標本の画像は正確に検出不能であるように目の領域を十分にぼかすことができる。また、非常に低い解像度の画像、例えば、 320×240 画面のネイル画像上で処理が生じると、赤目をまさに視覚的に見ることができないという可能性が非常に高い。

【0049】

50

検出が元の画像解像度で実行されると、赤目の座標及び半径が解像度非依存型の座標に保存される。これら座標と半径の双方によって、いかなる解像度の赤目も削除/除去することができる。この除去が非常に低い解像度、例えば、320×240画面のネイル、に適用された場合、除去領域は小さい。一方、非常に高い解像度の画像上で実行された場合、除去領域は赤目の座標を中心により大きくなる。

【0050】

人工照明調整フィルタもまた、2段階式アプローチを要求する操作である。第1のパスは、特定の光（例えば、白熱灯の照明用の黄色）がどの程度写真に存在するかを決定するために実行される。一定の割合（即ち、写真の25%）以上が特定の色範囲を含む場合、画像は修正される。上記決定がこの一定の割合に基づくので、正確な量が解像度によってわずかに変動することがある。例えば、厳密に画像の25%が元の画像解像度で黄色を含む場合がある。画像がダウンサンプリングされる時、計算された割合は通常では、25%よりわずかに少なかったり多かったりする。

10

【0051】

すべての解像度における一貫性に対し、色の検出は元の画像解像度において実行され、分析段階の間に記憶される。レンダリングが発生すると、フィルタ段階がいずれかの解像度で、但し、分析段階の間に元の画像解像度データから収集されたデータに基づいて実行される。元の画像データの解像度において分析段階を実行し、今後のレンダリングの結果を記憶させることによって、フィルタ段階がいずれかの解像度で実行可能である。この手法は、すべての解像度にわたって解像度非依存の動作をシミュレートする。

20

【0052】

内部パラメータの修正

第3のクラスの擬似解像度非依存の画像操作は、通常では、一定値又は画像の解像度の関数となるある範囲の値に基づいて、内部パラメータ値を修正する。例えば、リップルフィルタは、小石を写真上にレンダリングされた池に落とす結果をシミュレートする。ほとんどのアルゴリズムはこれを、画像全体に数学的波動関数をシミュレートすることによって実行する。一般的ではあるが、これは画像のサイズに依存する。このため、異なる数のリップルが、同じ画像の異なる解像度で認められる。

【0053】

代替のアプローチは、異なる解像度が同じように見える（即ち、同数のリップルを有する）ように画像の解像度をパラメータとして採用する数学的波動関数を有することである。これは、解像度に基づいて、パラメータを線形に、又はパラメータ変更のための対数関数あるいは幾何関数を決定して変更することによって可能であることもある。

30

【0054】

多重不連続解像度

第4のクラスの擬似解像度非依存の画像操作は、異なる解像度に対し異なる不連続の内部パラメータ値を選択する。これらの画像操作では、内部パラメータのそれぞれに対し数学関数を定義することは一般に不可能である。

【0055】

異なる解像度の画像に対し不連続パラメータの設定を選択することによって解像度非依存の動作をシミュレートすることが可能であるので、すべての解像度にわたって近似値が求められる。まず、画像の最大サイズが決定される。次に、その最大サイズが利用できる解像度のそれぞれと比較される。最も厳密に一致する解像度が選択される。

40

【0056】

ペイント効果は、チョーク/チャコール又はフレスコ・アート効果等の、種々の芸術的効果をシミュレートする画像操作の集合である。このような操作に対し、サポートされたサイズは、128、256、512、1024、2048、及び4086を含むことができる。128未満の画像については、128に対応付けられる設定が使用される。これらの数がどのように選択されるかについては特別なことはない。200、400、800、1600、及び3200等のサイズをサポートするために別の画像操作が選択される。所与

50

の画像操作では、一定した解像度の特定の集合がサポートされる。

【0057】

サポートされたサイズに正確に一致しない画像がレンダリングされると、2つのアプローチが現在使用される。最も一般的なアプローチは、レンダリングされている画像のサイズに最も近いパラメータ設定を選択することである。このアプローチは、このクラスにおける多くの画像操作に対しすべての解像度において一貫した結果を付与する。

【0058】

代替のアプローチは、画像操作によってサポートされた正確な中間サイズに一致するように画像をサイズ変更することである。ある状況において、これは、画像操作の正確なサポートされた解像度が所望の出力解像度に一致しないことと仮定して、異なる解像度においてより一貫性のあるレンダリングになるという結果になる。この操作が適用された後で、画像はもとの所望の出力解像度にサイズ変更される必要はない。画像操作の一部又はすべてが同じサイズをサポートする場合、これは画像の唯一の再サンプリングが必要とされるので、より望ましいアプローチである。残念なことに、これは、更なる再サンプリング操作によってもたらされるマイナスの効果を生じる。

10

【0059】

解像度依存操作

このカテゴリに分類される画像操作は、定義により非解像度非依存型である。より重要なことには、こうした操作は、擬似解像度非依存型の説明において記載されたように解像度非依存型に近似することはできない。

20

【0060】

サポートされる解像度依存の画像操作の数を最小限にする、あるいは解像度依存操作をサポートしないことが目標とされる。このことは、従来、アプリケーションが画像操作をほとんどサポートしない結果となるので、実行できるものではなかった。解像度の連続的又は不連続的集合において解像度非依存型をシミュレートできる（即ち、擬似解像度非依存型）ことにより、今ではアプリケーションが実行可能な選択肢として解像度依存の操作を単にサポートしないことが実現可能になった。

【0061】

レンダリング実施詳細

最良の形態において、レンダリング・パイプラインは、最も効果的な方法で画像操作のそれぞれをレンダリングする方法を決定するために、上記各画像操作のそれぞれの特徴を考慮に入れる。これらの特徴は、画像操作が解像度非依存型、擬似解像度非依存型、又はそのどちらでもない（即ち、解像度に依存した）かを判断する。また、種々の画像操作間の関係は、各操作が互いに影響し合うこともあるので、定義されなければならない。例えば、回転が後に続くチョーク/チャコールペイント効果のアプリケーションは、ペイント効果のアプリケーションの前に回転が適用された場合と比較して、異なる結果が生じることになる。

30

【0062】

すべての操作が解像度に依存していない場合、レンダリングパイプラインは、フラッシュピクス（商標）画像モデルによって定義されるものと非常に類似している。この場合、種々の画像操作間の関係は依然として特定されなければならないが、すべての操作は同じ解像度において起こる可能性がある。さらに重要なことには、すべての操作が低解像度画像データ上で実行可能であり、処理能力及びメモリがより少なくてすむ。

40

【0063】

画像処理操作の少なくとも1つが擬似解像度非依存の画像操作を含む場合、本発明の利点がより明白になる。この場合、レンダリングは所望の出力解像度に近いある所定の解像度において必要とされるが、アクセス又はレンダリングされる元の解像度における画像データを必要としない。

【0064】

本発明の基本パイプライン実施

50

図4を参照すると、本発明によって定義されるような画像をレンダリングするためのプロセスの詳細が示されている。プロセス400は402から開始し、特定のデジタル画像がアプリケーションによって開かれる。404では、デジタル画像の解像度が決定され、元の画像解像度(original Image Resolution)に記録される。406では、出力装置の解像度が決定され、変数最終出力解像度(final Output Resolution)がこの値に設定される。

【0065】

408では、プロセス500によって、初期の動作解像度が何であるべきかが決定され、現在の出力解像度(current Output Resolution)がこの値に設定される。410では、画像は適切なバッファに複写され、408で特定された解像度に一致するようにサイズ変更され、制御は412へ渡される。

10

【0066】

412において、画像編集操作が実行されるべきかどうか判断される。実行すべき画像操作がない場合、制御は450に渡される。そうでない場合、414において、画像操作が解像度非依存の操作であるかどうか判断される。解像度非依存の操作でなければ、制御は420に渡される(図4A参照)。そうでない場合、制御は416に渡され、画像操作が現在の出力解像度で実行される。制御はその後、412に戻される。

【0067】

420(図4A参照)では、この操作が擬似解像度非依存であるかどうか判断される。肯定されると、制御は422に渡される。そうでない場合、制御は440に渡されるが、これは解像度依存操作だからである。440において、画像操作に最適な解像度(best Resolution For Image Operation)が元の画像解像度に設定されるが、これは解像度依存操作だからである。制御はその後、442に渡される。

20

【0068】

422では、擬似解像度非依存の操作である。そこで、分析実行(Do Analyze)段階が要求されるかどうか判断される。要求される場合、424において分析実行段階が制御が426に渡された後にプロセス600によって処理される。分析実行段階が要求されない場合、制御は直接426に渡され、プロセス700によってこの擬似解像度非依存の画像操作に対し一番近くでサポートされる解像度が何であるかが判断され、画像操作に最適な解像度をこの値に設定する。制御はその後、442に渡される。

30

【0069】

442において、画像操作の最適な解像度が現在の出力解像度に一致する場合、制御は416(図4)に戻され、画像操作が画像上で実行される。解像度が一致しない場合、制御は444に渡され、画像が画像操作の最適な解像度に一致するようにサイズ変更される。446では、現在の出力解像度が画像操作に最適な解像度に設定される。制御はその後、416(図4)に戻され、画像操作が画像上で実行される。

【0070】

450(図4)では、すべての画像操作が実行され、現在の出力解像度と最終出力解像度のそれぞれの値が比較される。これらの値が等しい場合、制御は454へ渡され、処理された画像は出力表示装置に送信される。そうでない場合、制御は452に渡され、画像は、最終出力解像度に一致するようにサイズ変更/再サンプリングされる。制御は454へ渡され、処理された画像は出力表示装置に送信される。

40

【0071】

すべての画像操作が解像度非依存であり、いずれの解像度でも実行可能であれば、すべての操作が最終出力解像度で実行されることは、フローチャートから明らかである。しかしながら、画像操作が解像度依存である場合、画像の元の解像度(original Image Resolution)で実行されなければならない。すべての画像操作が実行された後に、画像は出力装置に渡される前に、また最終出力解像度へと再スケーリング/再サンプリングされなければならない。

【0072】

50

一般に、擬似解像度非依存の操作を処理する場合、再サンプリングが必要とされず（画像操作に最適な解像度と最終出力解像度は同一である）、又は、最終出力解像度により近い解像度が、解像度依存操作であるなら元の画像解像度を要求することと比較して使用される。

【0073】

初期動作解像度の決定（図5のプロセス500）

幾つかの異なるヒューリスティクス（発見的方法）は、所与の画像操作の理想的な解像度（画像操作に最適な解像度）を決定するために適用される。この値は当該画像操作に依存するので、一般的なアプローチは以下に記載される。

【0074】

一般規則では、可能であれば現在の出力解像度を変更しないことが最良だとされる。これは画像忠実度を維持する。理想的には、図4のステップ452で実行されたような再サンプリング/再スケーリングが不要であるように画像操作に最適な解像度が最終出力解像度と同一である。しかしながら、これは不可能であることもある。

【0075】

次のアプローチは、すべての画像操作にわたって不変であり有効である解像度を検出することである。このように、多くとも2つの再サンプル操作、即ち、元の画像解像度から目標とされた解像度（target Resolution）への再サンプル（この場合、すべての操作に対し同じである）と、出力装置によって要求される最終出力解像度への最終再サンプルを実行する必要がある。すべての画像操作が同じ解像度で動作できる場合、異なる再サンプリングは必要ではない。プロセス500及び700は、その計算の際にこれら規則及びアプローチを考慮に入れる。

【0076】

連続範囲モデルが実現可能でないとすれば、解像度の特定の不連続集合に対し解像度非依存型をサポートすることが擬似解像度非依存の画像操作の設計上の目的である。好ましい実施の形態において、この不連続集合はすべての擬似解像度非依存の操作に対し同じである。このようにして、必要とされる再サンプリングの量が低減される。もしこれが不可能であっても、擬似解像度非依存の画像操作が解像度依存操作よりも依然として好ましいとされるのは、レンダリングが通常、元の画像解像度で発生する必要がないからである。

【0077】

図5を参照し、プロセス500では、元の画像解像度及び最終出力解像度に基づいて、処理される一連の画像操作に対し1つの最適な解像度を選択するプロセスを詳細に説明する。技術的に、このプロセスはオプションである。プロセス500の一つの具体化は、最終出力解像度を「1つの最適な解像度」として単に返すことである。これが有効であるのは、プロセス700により、特定の画像操作の要求にこたえるように画像をサイズ変更する結果になるからである。

【0078】

しかしながら、結果として画像劣化が生じるために、発生する再サンプリングの量を制限することは有益である。さらに望ましいアプローチは、実行中のすべての画像操作に対し1つの最適な画像を日和見的に検出することである。これはいつでも可能ではないかもしれないが、好ましい実施の形態では、最適な初期解像度を検出するために試みられる。

【0079】

プロセス500は502から開始し、目標解像度が元の画像解像度及び最終出力解像度のいずれか小さい方に設定される。仮定として、元の画像解像度よりも低い解像度で操作を実行する時、最終出力解像度で操作を実行することは許可される。これは、元の画像解像度に相対して最終出力解像度で処理されても、操作が解像度非依存型又は擬似解像度非依存型であると見なされ、一貫した出力を生じる結果になるからである。しかしながら、操作が元の画像解像度よりも高い解像度で実行される場合、それを元の画像解像度で実行することは容認できる。これが性能とメモリのトレードオフであるのは、この場合に元の画像解像度で操作を実行し、その後、最終出力解像度に一致するように画像を再サンプリン

10

20

30

40

50

グすることが容認できると仮定されるからである。一部のシステム及び操作では、このようなことが望ましくないこともあり、これらの実施の形態において、目標解像度が502において最終出力解像度に設定される（高解像度処理に関する説明を参照）。

【0080】

504において、画像編集操作が実行されるべきかどうか判断される。実行すべき画像操作がないと、制御は550に渡される。そうでない場合、506では、すべての画像編集操作が解像度非依存型かどうか判断される。その判断が真であれば、制御は550へ渡される。そうでない場合、制御は510に渡される。

【0081】

510では、すべての画像操作が擬似解像度非依存型又は解像度非依存型であるかが判断される。その判断が偽であれば、制御は530へ渡される。そうでない場合、512において、すべての操作が目標解像度をサポートするかどうか判断される。その判断が真であれば、制御は550へ渡される。そうでない場合、制御は516へ渡される。

【0082】

516では、すべての擬似解像度非依存型画像操作が目標解像度に最も近いある「共通解像度」をサポートするかどうか判断される。各操作を検討することによって、目標解像度に最も近い1つの解像度がサポートされるかどうか判断される。これが真であれば、制御は518に渡され、目標解像度がこの「共通解像度」に設定され、制御は550に渡される。そうでない場合、制御は530に渡される。

【0083】

530では、共通解像度が決定されないか、あるいは操作が解像度に依存する。このため、目標解像度は元の画像解像度に設定されるので、元の画像解像度で処理が実行でき、これにより、すべての解像度において一貫した結果を保証する。制御は550に渡される。

【0084】

好ましい実施の形態において、ステップ530が存在しないのは、最終出力解像度に近いある共通の目標解像度がピクチャIQ画像システムにおいてすべてのサポートされた画像操作に対し常に決定されるからである。これが事実と異なる場合、擬似解像度非依存操作が解像度依存操作と同じように処理されることがある。以後、解像度非依存型又は擬似解像度非依存型の操作を最終出力解像度に近い解像度で最初に行われるように、操作の並べ替え等の更なる作業が実行されるだろう。

【0085】

550において、目標解像度が発信側に戻される。

【0086】

分析実行 (Do Analyze) 段階の処理 (図6のプロセス600)

図6を参照すると、本発明によって定義されるような分析実行段階を処理するためのプロセスの詳細を示すフローチャートが示される。プロセス600は602から開始し、以前の分析実行段階からの結果が存在し、依然として有効であるかどうか判断される。これが真であれば、制御は650に渡される。そうでない場合、604において、分析実行が現在の出力解像度で実行できるかどうか判断される。これが真であれば、606において分析実行が現在の出力解像度で実行され、制御は650に渡される。これが事実でなければ、分析実行を608において元の画像解像度で実行しなければならない。

【0087】

608において、現在の出力解像度における蓄積結果を元の画像解像度にサイズ変更することが十分であるかどうか、あるいは、すべての画像操作が元の画像解像度で元の画像に再適用されなければならないかどうか判断される。元の画像解像度におけるすべての操作の再処理が要求されると、制御は620に渡される。そうでない場合、610において、作業画像が別個のバッファに複製される。612において、作業画像が元の画像解像度にサイズ変更される。制御はその後606に渡され、分析実行が実行される。

【0088】

620において、元の画像解像度の元の画像が検索される。622では、分析実行データ

を必要とする現在の画像操作に基づいて、この操作までのシーケンスにおけるどの操作が元の解像度に適用されることになるかが判断される。これは、操作によって異なることがある。元の解像度画像が処理された後で、制御は606に渡され、分析実行が実行される。

【0089】

650において、分析実行段階が終了し、すべての内部変数が未だ存続され且つ保存される。制御は発信側に戻される。

【0090】

最近接のサポートされた解像度を決定する(図7のプロセス700)

図7を参照すると、プロセス700は、元の画像解像度及び現在の出力解像度に基づいて、処理中の所与の画像操作に対し最良の解像度(画像操作の最良解像度)を選択するプロセスの詳細が示される。プロセス700は702から開始し、現在の出力解像度と比較される現在の画像操作のための最近接のサポートされた解像度が決定され、画像操作の最良解像度がこの値に設定される。 10

【0091】

すべての操作に対し一貫した解像度を検出することはできない。好ましい実施の形態では、すべての擬似解像度依存操作によってサポートされる元の画像解像度よりも一般的にはるかに低い、近接した解像度を検出することは常に可能である。定義によれば、すべてのピクチャIQ画像操作はこのカテゴリーに分類される。

【0092】

一部の実施の形態において、こうしたことが不可能であることもある。これが発生する時、元の画像解像度を使用できる。しかしながら、操作が適用されるたびに画像を再サンプリングすることは一般に望ましいことではなく、この理由は結果として画像が劣化されるからである。 20

【0093】

別の実施の形態において、すべての画像操作に必要とされる2つ又は3つの通常要求される解像度が存在する場合、必要ならば種々の操作間でスケールリングするために更なる再サンプル操作のオーバーヘッドでそれぞれをサポートすることが望ましいこともある。これは、操作を並べ替える能力に基づいて行なわなければならないアプリケーションに特化したトレードオフである。 30

【0094】

解像度依存型画像操作の処理

利用される画像操作が解像度非依存型、また擬似解像度非依存型のいずれかである時に、本発明の最大の利点の実現されると同時に、他の利点もまだ可能である。例えば、実行されるすべての画像操作のシーケンスにおいて解像度依存型操作が後で(理想的には最後に)実行できるように、画像操作を並べ替えることが可能である。

【0095】

これを行うことによって、解像度非依存型か擬似解像度非依存型のいずれかである画像操作の多くは、低解像度、理想的には最終出力解像度又はそれに近い解像度で実行できる。それらの操作が最終出力解像度で実行された後に、画像は解像度依存操作の処理のための元の画像解像度に再サンプリングされなければならない。最後に、画像は最終出力解像度に戻って再サンプリングされなければならない。再サンプル操作に必要な潜在的品質上の損失又は特別な時間/資源がトレードオフに値するかどうかについてはアプリケーションに依存する。 40

【0096】

本発明の最も大きな利点のはるかに低解像度の画像データ上で画像操作を実行できることであると同時に、その他の利点は元の解像度に比べてより高い解像度の画像データを処理する時に実現される。

【0097】

一般的に、ほとんどのアプリケーションが元の画像解像度の画像操作を適用し、それによ 50

ってその画像を縮小又は拡大するようにサイズ変更し、出力装置に一致させる。高解像度印刷装置に出力する時、プリンタの解像度は元の画像解像度よりも一般にははるかに高い。この場合、すべての操作が適用された元の画像は、プリンタに一致するようにより高い解像度にサイズ変更/再サンプリングされる。

【0098】

このアプローチでの欠点は、より高い解像度への再サンプリングが、画素化 (blockiness、濃淡のむら) 及び他のエイリアシング効果を含む、その他のアーチファクトを生じさせることである。これは、既に存在しないためにより高い解像度の画像データが「生成」されなければならないからである。この問題を解決するために、(画素化が最も一般に普及されている) 画像の高周波は、エイリアシングの効果を軽減するためにわずかに不鮮明にされる。

10

【0099】

エッジを減衰させる画像処理操作 (例えば、シャープネス操作又は強調されたエッジ) が実行されると、再サンプリング操作が濃淡むらの発生や、ぼかしの程度のいずれかの効果を有することがある。これらはそれぞれ、望ましくない。

【0100】

本発明によって実行できる代替のアプローチは、画像操作を適用することなく、出力装置の解像度に一致するように画像を最初に再サンプリングすることである。次に、種々の画像非依存型又は擬似解像度非依存型の操作が (元の画像解像度よりも高いと推定できる) 高解像度で実行される。これは、一貫しない結果を招くことから、解像度依存の操作では不可能である。

20

【0101】

一般に、このアプローチが使用される場合は注意しなければならない。これにより、各画像処理操作の処理時間及びメモリ要件が増加されるが、これは非常に高い解像度の画像データで実行されるからである。また、操作が解像度に依存する場合、元の画像解像度で実行されなければならない。

【0102】

さらに可能性の高いシナリオは、すべての解像度依存の操作を含む、元の画像解像度での画像操作の一部を処理する結果になるハイブリッドアプローチを使用することである。画像は再サンプリングされ、再サンプリング/サイズ変更操作の結果として望ましくないアーチファクトを示す解像度非依存型又は擬似解像度非依存型の操作の部分集合が、より高い解像度で実行される。これは、一部の操作が並べ替え可能であることを仮定する。

30

【0103】

本発明はすべての解像度にわたって一貫した結果を付与するために設計されている。既に述べたように、画像Aでは、画像操作が特定の解像度の画像に適用され、それより低い解像度にサイズ変更される。画像Bでは、画像が最初により低い解像度にサイズ変更され、画像操作が適用される。画像A及び画像Bが、再サンプリング/サイズ変更操作の間に生じられたエラーを考慮に入れて十分に接近している場合、その操作は解像度非依存型であると考えられる。言い換えると、ユーザが画像Aと画像Bを並列して見る場合、それらは視覚的に同じように見えることになる。

40

【0104】

理論上、この仮定が適用できる場合、解像度非依存型の影響が得られる。解像度非依存型又は擬似解像度非依存型の操作の一部の制限された集合では、全体に正しくないこともある。ある芸術的な効果はこの特徴を示している。この効果が元の解像度画像に適用される時、目に見える。しかしながら、既にその効果が適用された元の解像度画像が画面上のネイル又はサムネイルに再サンプリングされる時、あまりよく見えなかったり、あるいは完全に見えなくなる。

【0105】

画像処理の観点から、画像をより小さい解像度に再サンプリングする効果は、画像処理操作の効果を低減させる。例えば、シャープネス (シアープン) 操作が元の解像度に適用さ

50

れ、その結果がより小さい解像度に再サンプリングされると、シャープネスの量は低解像度ではより少なく見える。これは、再サンプリング操作がシャープエッジをぼかす傾向があり、シャープネス操作の効果を低減させるからである。

【0106】

このセクションは、解像度非依存型及び擬似解像度非依存型の両操作から、種々の異なる解像度で同じ操作を実行する時に解像度非依存の手法を使用することによる結果を示す出力を示す。図8は異なる解像度の元の画像を示している。

【0107】

各セクションでは4つの解像度、即ち、 100×67 、 200×134 、 400×286 、及び 800×536 が示される。操作は各解像度ごとに適用される。 800×536 の画像が、その解像度に適用される操作を有し、その後、この文書に適合するように再サンプリングされることに留意すべきである。これにより、 800×536 及び 400×286 の各解像度の直接比較を可能にする。理論上、純粋な解像度非依存操作は、出力が同じ表示サイズで並列比較される時、同一結果をもたらすことがある。

10

【0108】

図9及び図10は、同一フィルタの擬似解像度非依存バージョン及び解像度依存バージョンについてのリップルフィルタの出力を示している。リップル効果の画面状態が解像度依存出力に対して示される図中の解像度と共にどのように変化するか留意すべきである。これは、アルゴリズムに対する内部パラメータが解像度に基づいて修正されるフィルタの一例である。

20

【0109】

図11及び図12は、両バージョンに対するモザイクタイルフィルタの出力を示している。解像度依存の出力に対し示される図中の解像度に基づいてタイルの数がどのように変化するか留意すべきである。これは、アルゴリズムに対する内部パラメータが解像度に基づいて修正される一例である。

【0110】

図13及び図14は、フレスコ画法効果に対する出力を示している。解像度依存の出力に対する種々の解像度間の大きな相違に留意すべきである。これは、解像度の不連続集合がサポートされる一例である。鋭い目に対し、解像度非依存の出力に対し示される図の異なる解像度間において多少相違が見られることもあるが、非常に類似している。明確には、これは解像度依存の出力よりもはるかに良い状態である。

30

【0111】

ここで留意すべき重要なことは、今日の技術(本発明以外の)を用いたアプリケーションがすべての解像度にわたって同じ結果をもたらそうとする場合、元の画像解像度で処理が実行されなければならないということである。フレスコ効果では、 1600×1072 の場合の元の画像は、 750 MHz のインテル社ペンティアム(登録商標)IIIで動作する解像度での処理を完了させるのに11秒かかるが、擬似解像度非依存手法を用いて 400×286 以下の画像の同じレンダリングに対しては1秒未満である。

【0112】

図15は、擬似解像度非依存及び解像度依存の両手法を用いてチョーク/チャコール画法効果の出力からの並列した結果を示している。これは、解像度の不連続集合がサポートされる一例である。比較のために、これら手法がすべての解像度において同じ結果に近似させようとすることから、同じサイズの出力を並列して見ることが有用であることもある。画素レベルにおいての比較のための最良の方法は、解像度に関係なく同じサイズで各画像を見ることである。

40

【0113】

この比較により、同じサイズで見られる時に異なる解像度がどのように比較されるかが示される。この場合、 200×134 は100%で見られるが、 400×268 は50%で見られ、 800×536 は25%で見られる。このようにして見る場合、どのくらいの近さで解像度が並列比較するかについて更に正確に決定されることができる。

50

【0114】

チョーク/チャコール画法では、擬似解像度非依存の手法を用いて解像度間に更なる相違が目に見える。これについての主な理由は、低解像度で動作する時に、一定量のデータが既に失われ、近似されなければならないという事実による。画面のネイル(320×240)まで再サンプリングされ、元の画像と比較するために拡大(ズームイン)される高解像度写真に対して同じ状況が存在する。明確には、図示されたものと同様のデータ損失がある。これは予期されることがある。これはまさに、ある画像操作が低解像度でより良く実行可能であり、この解像度を「より良く生成する」ことができるということである。

【0115】

本発明がデジタル画像システム(ビデオ又はスチル)とともに使用されるように説明してきたが、本発明を概して好適なデジタル画像システム上で実施されてもよいことは理解されるべきである。これは、直接エンドユーザ・ピアトゥピア接続システムと同様、PCベースの画像アプリケーション、共有、分配、又はセントラルサーバとクライアントとの間における画像データの表示を可能とするウェブ共有アプリケーションを含む。

10

【0116】

内蔵された情報機器又はデジタル画像装置の一部として含まれることができ、有線ネットワーク環境と無線ネットワーク環境との双方において同じようにうまく動作することができる。

【0117】

本発明の好ましい実施の形態によって画像データをレンダリングして擬似解像度非依存の動作をシミュレートするシステム及び方法の開示は、本質的には具体例にすぎず、本発明や、そのアプリケーション、あるいは使用法を決して限定するものではない。更に、上記の説明において、本発明の開示を更に十分に理解するために実施のための多数の特定された詳細が示される。しかしながら、本発明がこれらの特定の詳細なしでも実施できることは当業者には理解されるだろう。他の例において、本発明をあいまいにさせないために、周知の処理の特徴及び機能についての説明は省略されている。

20

【図面の簡単な説明】

【図1】

図1A及び図1Bは、2つの従来の画像処理の方法を詳述するフローチャートである。

【図2】

フラッシュピクス(商標)がデジタル画像を処理する方法を示すフローチャートである。

30

【図3】

デジタル画像が一定の解像度で処理される方法を示すフローチャートである。

【図4】

図4及び図4Aは、本発明が解像度依存画像をレンダリングする方法を示すフローチャートである。

【図5】

処理すべき一連の画像操作に対する最良の解像度を選択するための詳細を示すフローチャートである。

40

【図6】

本発明の「Do Analyze(分析実行)」という語句を処理するための実施を示すフローチャートである。

【図7】

元の解像度と現在の出力解像度に基づいて処理される画像の最良の解像度を選択するための処理を示すフローチャートである。

【図8】

異なる解像度のデジタル画像を示す図である。

【図9】

リップル効果のある解像度に好適な1つの解像度に適用させるために使用されるフィルタ

50

を示す図である。

【図10】

補正フィルタ動作をリップル効果に適用しない場合の結果を示す図である。

【図11】

タイル画像を効果的に解像度に依存せずにレンダリングするために使用されるフィルタを示す図である。

【図12】

補正フィルター動作なしに異なる解像度のタイルを示す図である。

【図13】

補正解像度フィルタを使用するフレスコ効果を示す図である。

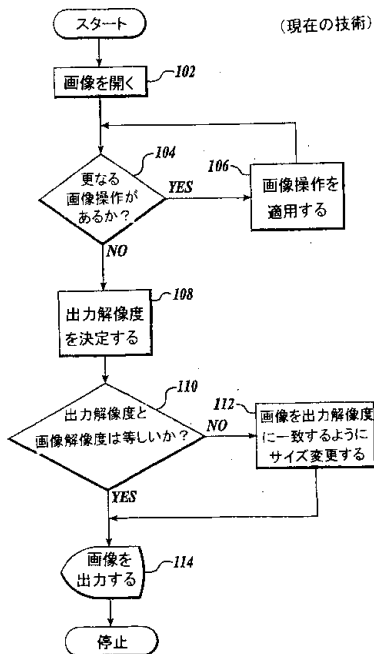
【図14】

補正フィルタ動作なしに同じフレスコ効果を示す図である。

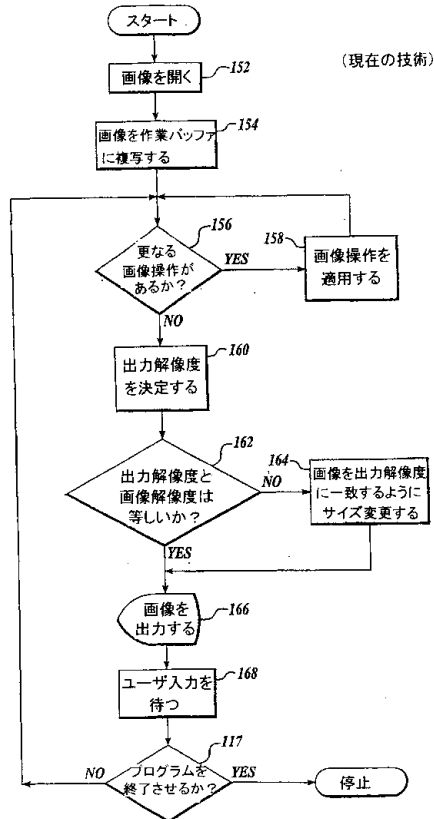
【図15】

補正フィルタ動作を使用するが、チョーク/チャコール画像効果上ではそのフィルタ動作を使用しない並列比較を示す図である。

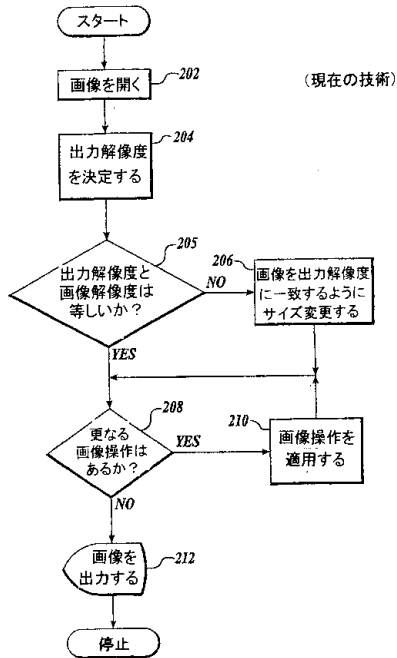
【図1A】



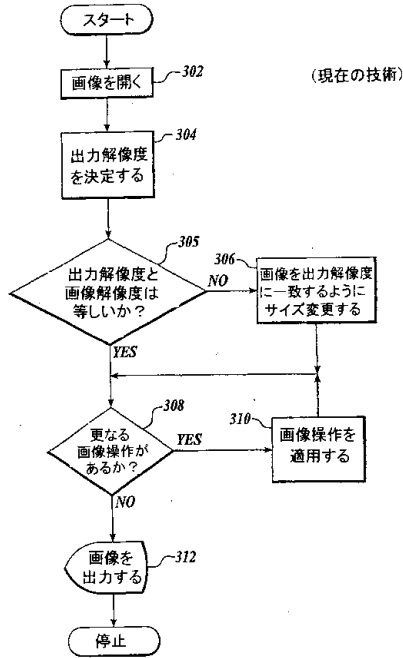
【図1B】



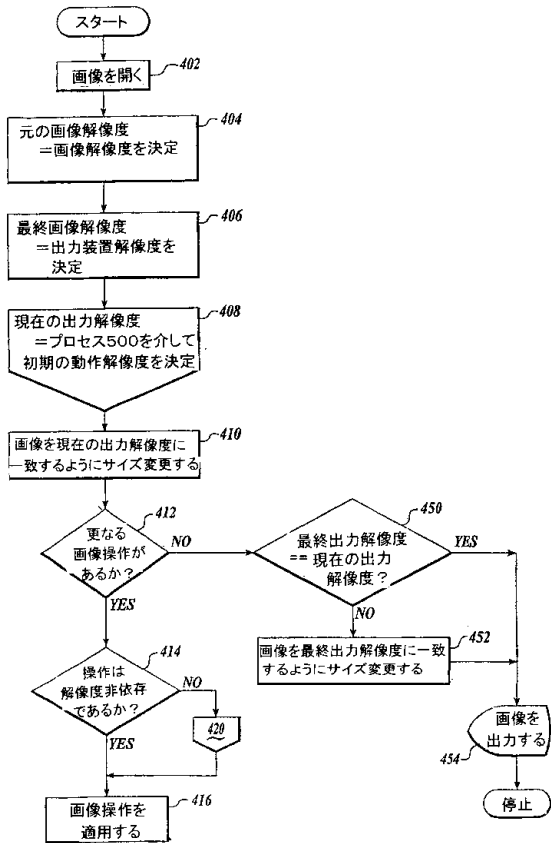
【 図 2 】



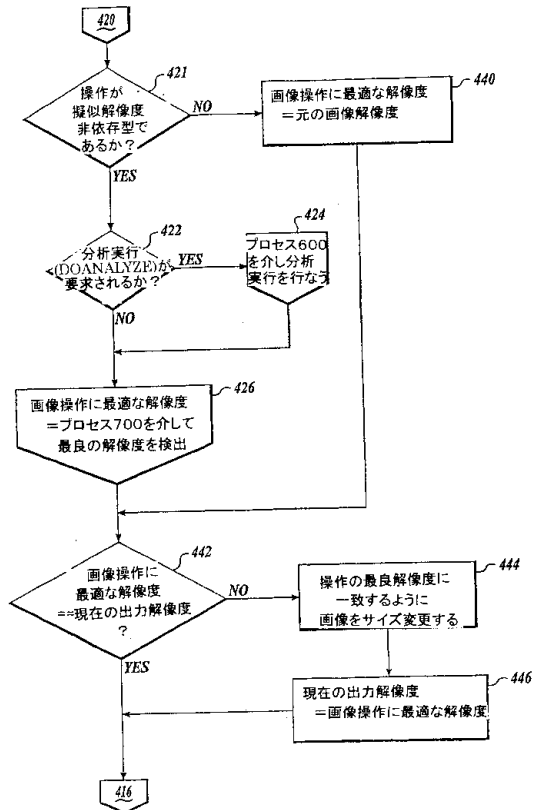
【 図 3 】



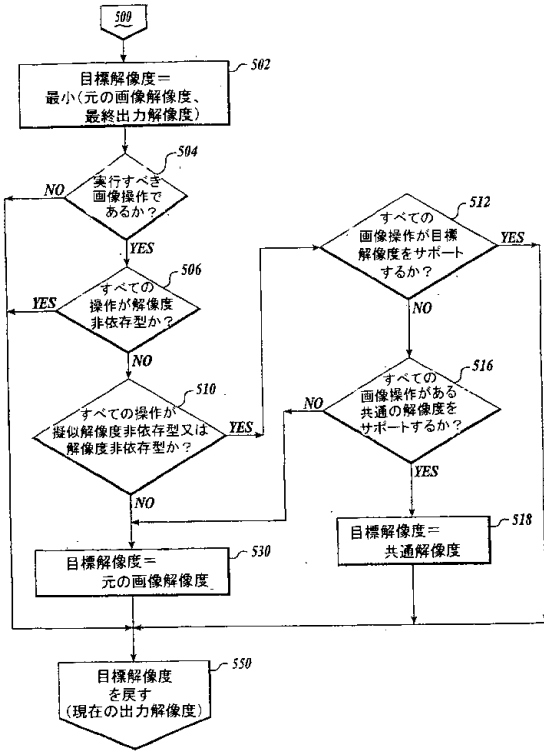
【 図 4 】



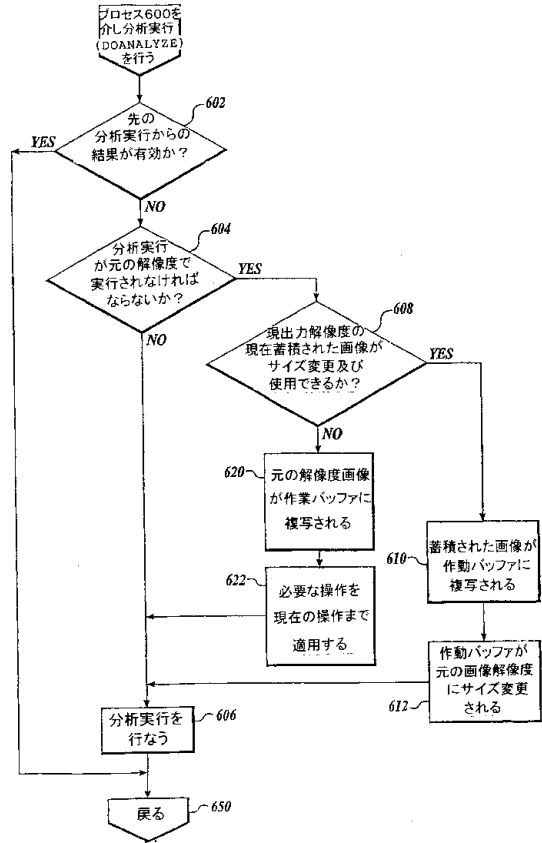
【 図 4 A 】



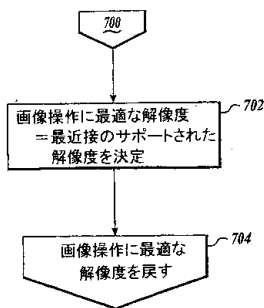
【 図 5 】



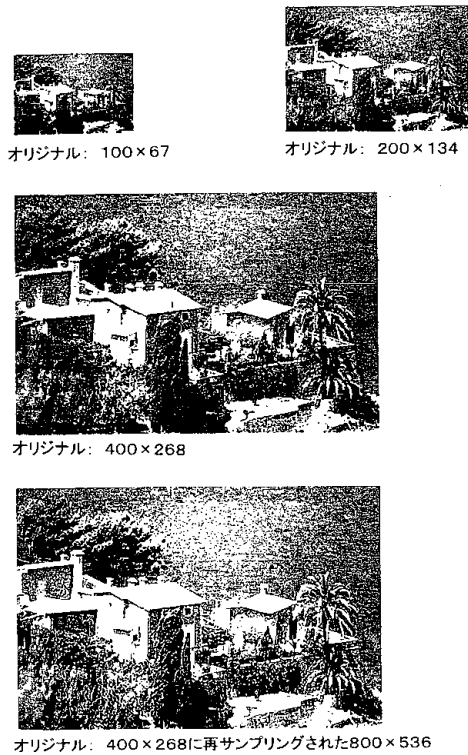
【 図 6 】



【 図 7 】



【 図 8 】



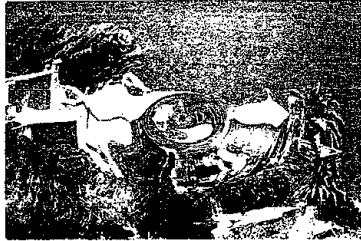
【 図 9 】



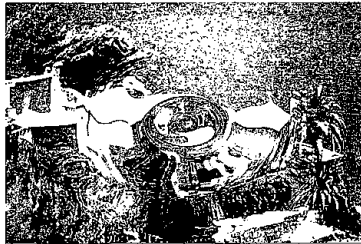
解像度非依存リップル: 100×67



解像度非依存リップル: 200×134



解像度非依存リップル: 400×268



解像度非依存リップル: 800×536に適用され400×268に再サンプリングされる

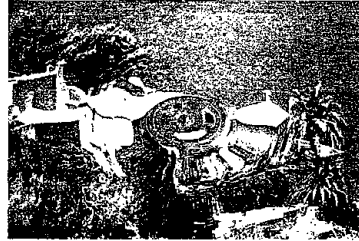
【 図 10 】



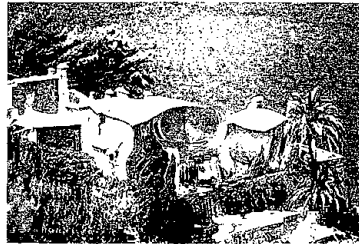
解像度依存リップル: 100×67



解像度依存リップル: 200×134



解像度依存リップル: 400×268

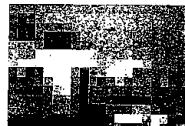


解像度依存リップル: 800×536に適用され400×268に再サンプリングされる

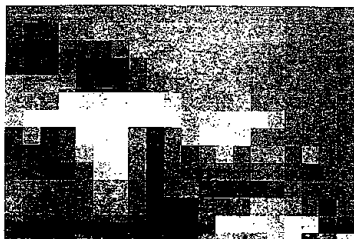
【 図 11 】



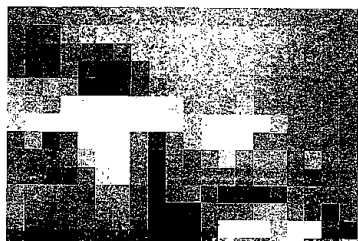
解像度非依存モザイク: 100×67



解像度非依存モザイク: 200×134



解像度非依存モザイク: 400×268



解像度非依存モザイク: 800×536に適用され400×268に再サンプリングされる

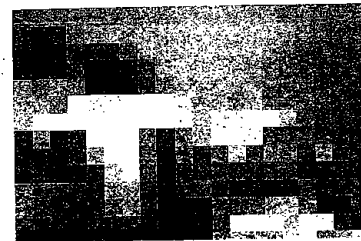
【 図 12 】



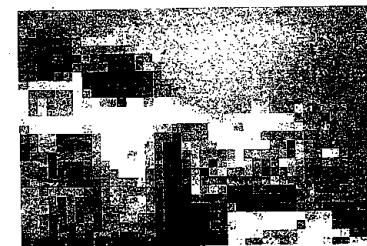
解像度依存モザイク: 100×67



解像度依存モザイク: 200×134



解像度依存モザイク: 400×268

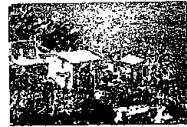


解像度依存モザイク: 800×536に適用され400×268に再サンプリングされる

【図 13】



解像度非依存フレスコ: 100×67



解像度非依存フレスコ: 200×134



解像度非依存フレスコ: 400×268



解像度非依存フレスコ: 800×536に適用され400×268に再サンプリングされる

【図 14】



解像度依存フレスコ: 100×67



解像度依存フレスコ: 200×134



解像度依存フレスコ: 400×268



解像度依存フレスコ: 800×536に適用され400×268に再サンプリングされる

【図 15】

解像度非依存型アプローチ

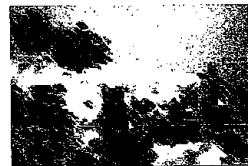


100×67に適用

解像度依存型アプローチ



200×134の画像に適用



400×268の画像に適用され200×134に再サンプリングされる



800×536の画像に適用され200×134に再サンプリングされる



【国際公開パンフレット】

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
18 April 2002 (18.04.2002)

PCT

(10) International Publication Number
WO 02/31754 A1

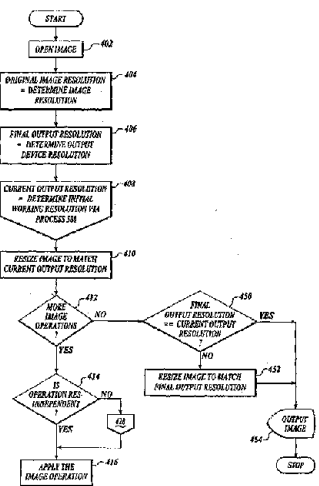
- (51) International Patent Classification: G06K 9/32
- (74) Agent: DAVISON, James, Northwest Patents, 19822
256th Avenue NE, Woodinville, WA 98072 (US).
- (21) International Application Number: PCT/US01/42694
- (81) Designated States (national): JP, US.
- (22) International Filing Date: 12 October 2001 (12.10.2001)
- (84) Designated States (regional): European patent (AT, BE,
CZ, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC,
NL, PT, SE, SK).
- (25) Filing Language: English
- (85) Published:
— with international search report
— before the expiration of the time limit for amending the
claims and to be republished in the event of receipt of
amendments
- (26) Publication Language: English
- (30) Priority Data: 13 October 2000 (US,10,2000) US
60/240,495
- (71) Applicant (for all designated States except US): PIC-
TUREIQ INC. (US/US); IP COUNSEL, 505 136th Ave
South Suite 350, Seattle, WA 98104 (US).
- (72) Inventor; and
- (75) Inventor/Applicant (for US only): WILKINS, David
[US/US], 15 Chen Drive, Providence, RI 02906 (US).

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: TECHNIQUES FOR RESOLUTION-INDEPENDENT RENDERING OF IMAGES



WO 02/31754 A1



(57) Abstract: A method to improve the performance of rendering image data (402) by converting what would normally be considered resolution-dependent image behavior into behavior that is substantially resolution independent. This allows significant performance improvement since the rendering (454) can be performed on the lower-resolution image data used, for example, for on screen viewing and when the image effect is applied to a higher resolution rendering, the effect, as viewed, is substantially the same as the effect viewed at a lower resolution. This conversion of normally resolution-dependent behaviors into pseudo-resolution-independent behaviors also allows the image effects to be applied to be carried out on a lower resolution image with confidence that when the image is rendered at a higher resolution that the image effects applied will substantially have the same appearance that the effect had at the lower resolution.

WO 02/31754

PCT/US01/42694

TECHNIQUES FOR RESOLUTION-INDEPENDENT RENDERING OF IMAGES

5 This application is entitled to the benefit of Provisional Patent Application Serial
Number 60/240,495 filed October 13, 2000.

BACKGROUND OF THE INVENTION

10 1. Field of Invention

The present invention relates generally to digital image processing systems. More particularly the invention describes techniques that greatly improve the performance of rendering image data. More particularly, techniques are described that take
15 advantage of resolution-independent characteristics, even for operations that are traditionally not considered to be resolution-independent. More specifically, certain techniques are applied to simulate resolution-independent behavior (called *pseudo-resolution-independent*) when an image operation is applied at any resolution.

20 These techniques can greatly improve the performance of rendering systems in computer applications, such as those executing on personal computers, digital imaging consumer appliance devices, and when viewing and manipulating photos over a network environment (in the form of client-side or the server-side executable code on either a physically wired or wireless medium).

25 Most image operations fall into two categories: those that generate the same result, regardless of the resolution of the image (i.e. *resolution-independent*) and those that generate significantly different visible results for different resolutions of image data (i.e. *resolution-dependent*). Rotation, for example, is invariant across all resolutions and is *resolution-independent*. However, a special effects image processing
30 operation such as a Fresco paint effect varies widely across different resolutions and is *resolution-dependent*.

WO 02/31754

PCT/US01/42694

If only resolution-independent operations are supported, a significant performance improvement is achieved since rendering can be performed on the lower-resolution image data that is needed for on-screen viewing of the output. Because, in general, the resolution of the display is usually much less than that of most other output devices, such as a high-resolution printer.

If all, or at least the majority, of the resolution-dependent image operations behave in such a way that they simulate resolution-independent behavior, most of the benefits of resolution-independence are still realized. While it may not be possible to support a continuous range of resolutions in a resolution-independent manner, certain assumptions and constraints are made that allows resolution-independence to be approximated, thus creating a new class of pseudo-resolution-independent image operations. In the preferred embodiment, all operations are either resolution-independent or pseudo-resolution-independent.

One of the objects of this invention is the ability to efficiently process image data at lower-resolutions, but then also be able to obtain consistent results across all resolutions. This is particularly important when viewing and processing images in on-screen display devices, since in general these devices are much lower resolution than the resolution of the image data. Moreover, this invention provides this efficiency without sacrificing consistent results across all resolution, something most applications today cannot correctly support.

Another object of the invention is to be able to process and transmit much less image data than would otherwise be necessary, which is essential in a network-connected environment. If the client must download image data, where it is processed locally (on the client), it is of much benefit to only require that a small amount of data be downloaded. Today, the original resolution image must be transferred to the local client so that the image operations can achieve consistent results even though a small screen resolution view (less than the original resolution image) is needed. Even when the operations are performed on the server, the ability to perform the image operations at the screen resolution of the image data significantly reduces the processing and memory requirements of the server, thus permitting many more transactions to be performed in the same amount of time.

WO 02/31754

PCT/US01/42694

An equally important benefit is that these techniques allow digital imaging operations to be performed on low-cost consumer electronics imaging devices that have low-processing power and little memory. Up until now, this has been difficult to achieve
5 due to the system constraints.

2. Description of Relevant Art

Traditional image processing applications, such as Adobe Photoshop™, assume that
10 the best way to get consistent results for all image processing operations is to perform the operations at the original image resolution, regardless of the targeted output device's resolution (whether it is higher or lower than the original image resolution). This guarantees consistent results, regardless if the image is viewed on a low-resolution video display or a high-resolution printer. Traditional image
15 processing applications usually support one of two approaches listed below:

The first approach, shown in Figure 1a, accumulates the sequence of image operations that are applied and re-renders each operation when an updated output image is requested from the output device. This involves the complete reprocessing
20 of all image operations at the image's original resolution. This can be a very slow process since each operation is reapplied at the original image resolution. This gives the user the flexibility to have unlimited undo capabilities since each operation is stored in a list of operations.

25 Referring now to Figure 1a, process 100 is a flowchart that details the processing of a digital image using a traditional image-editing model. Process 100 begins at 102 when a particular digital image is opened by the application at the resolution of the image. There is no resampling/rescaling that occurs. At 104, a determination is made if any image editing operations should be performed. If no image operations
30 are to be performed, control is pass to 108. Otherwise, at 106 the image operation is performed on the original image resolution data, then control is passed back to 104, where it is determined if there are additional image operations to be processed. It

WO 02/31754

PCT/US01/42694

should be noted, in this model successive image operation are applied on top of (or accumulated on) the previous image operations that have been applied to the image.

At 108, the desired output resolution is determined. At 110, if the original image resolution does not match the desired output resolution, then at 112 the image is copied into a temporary buffer and is resized to match the desired output resolution and then control is passed to 114. Otherwise, control is passed directly to 114 where the image is sent to the output device. When the output device selected is a display, the output resolution may vary depending if the image is enlarged or reduced (zoomed in/out) on the display. However, the key observation is that the image operations are generally applied to the original image resolution data, and after all operations are applied, the image is resized to match the resolution of the output device.

The second approach, shown in Figure 1b, performs each operation on the image data at the original image resolution, but in an intermediate working or cached buffer. This working buffer contains the entire image at the original image resolution. As each new operation is applied, it is performed directly on the working buffer. When a desired output resolution image is to be generated, the working buffer is resampled to match the desired output resolution. Clearly, the benefit is that each operation is normally applied once in the working buffer and does not need to be reprocessed when a new output resolution is requested. The disadvantage with this approach is that while unlimited undo/redo operations are possible, it is more difficult to support such a feature since the entire list of image operations are not normally re-rendered when an updated output image is requested.

Referring now to Figure 1b, a process 150 is a flowchart that details the processing of a digital image using this model. The process 150 begins at 152 when a particular digital image is opened by the application. At 154, the digital image is copied into the working buffer, at the same resolution as the original. At 156, a determination is made if any image editing operations should be performed. If no image operations are to be performed, control is pass to 160. Otherwise, at 158 the image operation is performed on the working buffer (with the accumulated operations), then control is

WO 02/31754

PCT/US01/42694

passed back to 156, where it is determined if there are additional image operations to be processed.

At 160, the desired output resolution is determined. At 162, if the resolution of the working buffer (which is the resolution of the original image) does not match the desired output resolution, then at 164 the working image is copied into a temporary buffer and is resized to match the desired output resolution. In either case, at 166 the resized image from the temporary buffer is displayed on the output device. The process then waits at 168 until either a request to re-render the image or a user request for a new image operation is applied. At 170, it is determined if the program should terminate, if so the appropriate steps are taken and the process stops. Otherwise, control is pass back to 156 to process the request. In this way, the entire list of image operations need not be reapplied to the working image buffer.

The approach taken by Adobe Photoshop™ is a combination of the two approaches. In this case, an intermediate working buffer (s) is used, but is considered a cache that can be flushed if the user performs an undo operation. The working buffer can be discarded and be regenerated based on the updated list of image operations. Alternatively, the working buffer can be reverted to a previous cached copy of the working buffer. Taken to an extreme, all intermediate operations can be cached, simulating the effect of infinite undo/redo operations. This is at the expense of using a significant amount of memory and/or disk space

While each of these approaches has the benefit of providing consistent results across all resolutions, it is generally slower since the image processing operation(s) must be performed on all the pixels at the original image resolution. Although using intermediate working buffer caches minimizes the need to reprocess all image operations each time a new render resolution is requested, the image operation(s) must still be performed on the original resolution the first time. Even on the fastest processors today, this can still be a time consuming operation.

It should be noted that variants of this model are possible when certain common resolution-independent operations are performed, such as rotate or color adjustment

WO 02/31754

PCT/US01/42694

filters. In these situations, it is common to resize/resample the working buffer to a much lower resolution, such as the screen resolution, where a preview of the operation can be shown. Using this technique, real-time manipulation is feasible for the limited set of resolution-independent operations. Once the user is satisfied with the results, the operation is then applied to the original resolution working image buffer.

While this does mitigate some of the disadvantages described above, only a small subset of image operations conform to this resolution-independent behavior or can support the desired real-time manipulation functionality.

More recently, new technology is available that allows efficient image rendering when all image operations are performed in a resolution-independent manner. An imaging application that supports FlashPix™ technology, developed by a consortium of companies comprising of the Digital Imaging Group (DIG), can display and manipulate on-screen images at a lower-resolution in a consistent manner, but retains the ability to process higher-resolution image data.

FlashPix™ technology defines a limited set of image operations including: rotation, translation, cropping, color twist, blur/sharpen adjustment, and brightness/contrast adjustment. Each of these operations is defined such that they can be performed at specific "powers of two" resolutions as defined by the FlashPix™ standard, but provides consistent results across all these resolutions.

Referring now to Figure 2, process 200 is a flowchart that details the processing of a digital image using the FlashPix™ imaging model. The process 200 begins at 202 when a particular digital image is opened by the application. At 204, the desired output resolution is determined. At 205, it is determined if the output resolution and the original image resolution are the same. If they are not, at 206, the image is resized to match the resolution of the output device. In either case, at 208, a determination is made if any image editing operations should be performed. If no image operations are to be performed, control is pass to 212. Otherwise, at 210 the image operation is performed on the image data resized in step 206. Control is then

WO 02/31754

PCT/US01/42694

passed back to 208, where it is determined if there is additional image operations to be performed.

At 212, the processed image is sent to the output display device. Since at 204 the
5 output resolution was already determined and all processing was performed on the
targeted output resolution, no additional resampling of the image is needed. It
should be noted, a similar working buffer cache could also be employed in the
FlashPix™ rendering model, as described in section 2.1 above.

10 The FlashPix™ imaging model provides the benefit of quick and efficient processing
of the image operations when performed on lower-resolution image data. This
provides for real-time manipulation and rendering on a low-resolution output display
device. While this is clearly beneficial, only a small set of resolution-independent
operations are actually supported. More particularly, it would be impossible to
15 support any operation that is resolution-dependent, or even pseudo-resolution-
independent since the architecture does not permit for such provisions as described
in this paper. It is also noted that FlashPix™ only supports specific "powers of two"
resolutions, and not a continuous range of resolutions as provided by this invention.

20 A completely opposite and simplistic approach is also used in which the resolution
requirements of the image operations are ignored and each operation is processed
at some fixed resolution. This resolution usually does not match the original image
resolution, nor does it provide for consistent behavior if the operations were applied
at different resolutions. For the cases when the image operations are resolution-
25 independent, as for those supported by FlashPix™, this model works and provides
consistent results. However, for those cases where the image operations are not
resolution-independent, more inconsistent results are seen. The degree of
inconsistency is a function of the specific image operation.

30 For example, when applying a resolution-dependent blur operation, such as a low-
pass 5x5 kernel, the amount of blur becomes less pronounced as the resolution (and
image size) increases. This may not be too objectionable since in both cases, the
image appears with some degree of blurriness. However, for a "ripple" effect, in

WO 02/31754

PCT/US01/42694

which a simulation of a ripple of a stone dropping into a lake is rendered onto an image, the actual number of concentric ripples and appearance varies depending on the resolution and the exact characteristics of the algorithm (see sample output for the ripple operation in section 5.2 and 5.3).

5

Referring now to Figure 3, process 300 is a flowchart that details the processing of a digital image using this approach. The process 300 begins at 302 when a particular digital image is opened by the application. At 304, the desired output resolution is determined. At 305, it is determined if the output resolution and the original image resolution are the same. If they are not, at 306, the image is resized to match the resolution of the output device. In either case, at 308, a determination is made if any image editing operations should be performed. If no image operations are to be performed, control is pass to 312. Otherwise, at 310 the image operation is performed on the image data resized in step 306. This occurs even though the image operation may require the original image resolution data; therefore, proper results might not be obtained. Control is then passed back to 308, where it is determined if additional image operations are to be performed.

At 312, the processed image is sent to the output display device. Since at 304 the output resolution was already determined and all processing was performed on the targeted output resolution, no additional resampling of the image is needed. It should be noted, a similar working buffer cache could also be employed in this rendering model, as described in section 2.1.

It is common for this type of approach to be taken on Web sites that support image processing operations, but where the primary output target is a low-resolution output display device. In general, after the image operation is applied to the image (at low-resolution), the results are commonly e-mailed to a recipient. The recipient then views the low-resolution result (already rasterized) on their low-resolution output display device.

For cases when the operation is reapplied to the original resolution image data for printing, such as through an on-line print fulfillment service, the situation is even

WO 02/31754

PCT/US01/42694

more problematic. The user may preview the output via a low-resolution rendered image on-screen. In this case, the output viewed on the display may not match the final printed output that is ordered.

5 Even if 80% of the time this discrepancy is not detected, for those that do notice the difference, they may request a refund of a printed output. This situation is very costly and is clearly not an acceptable solution. This invention creates a more desirable result since consistency is achieved across all resolutions for all operations, thus improving customer satisfaction.

10

More importantly, a photo service that takes the "ignore it" approach will most likely limit the supported image operations to minimize this problem. What is desired is the ability to offer a larger range of image operations, assuming these image operations are either resolution-independent or are structured that they can take advantage of resolution independent characteristics (i.e. pseudo-resolution-independent).

15

SUMMARY OF THE INVENTION

In summary, the present invention consists of a method for applying normally resolution-dependent image effects in such a manner that the effects have substantially the same appearance regardless of the final resolution of the image. This allows the effects to be applied to any image resolution with the confidence that regardless of the later resolution rendered, the resulting effect will have the same appearance as it had when it was applied to the original resolution. By determining which particular effect parameters are resolution dependent and then modifying those parameters with the modification values being predicated on the final image resolution being rendered, it is possible to convert resolution-dependent parameters into substantially resolution-independent parameters.

20
25
30

BRIEF DESCRIPTION OF THE DRAWINGS

WO 02/31754

PCT/US01/42694

The invention, together with further advantages thereof, may be best understood by reference to the following description taken in conjunction with the accompanying drawings in which:

- 5 Figures 1a and 1b show a flowchart that details two traditional methods of image processing.
- Figure 2 shows how FlashPix™ processes a digital image.
- Figure 3 shows how a digital image may be processed at a fixed resolution.
- Figure 4 and 4a are flowcharts that show how the present invention renders a resolution dependent image.
- 10 Figure 5 shows the details for selecting the best resolution for a series of image operations that are to be processed.
- Figure 6 shows an implementation for handling the "DoAnalyze" phase of the present invention.
- Figure 7 shows the process for selecting the best resolution for the image to be processed based on the original resolution and the current output resolution.
- 15 Figure 8 shows a digital image at different resolutions.
- Figure 9 shows a filter used to make the ripple effect applied to one resolution suitable for any resolution.
- Figure 10 shows the result of not applying a compensating filter action to the ripple effect.
- 20 Figure 11 shows a filter used to render the tiles image effectively resolution independent.
- Figure 12 shows the tiles at different resolutions without the compensating filter action.
- 25 Figure 13 shows a Fresco effect using a compensating resolution filter.
- Figure 14 shows the same Fresco effect without the compensating filter action.
- Figure 15 shows a side-by-side comparison of using a compensating filter action and not using the filter action on a chalk/charcoal image effect.

30

WO 02/31754

PCT/US01/42694

DETAILED DESCRIPTION OF THE EMBODIMENTS

The present invention was originally targeted for deployment on image processing servers, as part of an ASP (Application Service Provider) model that are deployed on systems handling between hundreds to thousands of simultaneous operations at one time. What is desired is a solution that is scalable across many computers. More importantly, it provides consistent results across all image resolutions regardless of the size of the image rendered, but with a decrease in rendering time as the output resolution decreases.

10

It should be noted that image data could range from between low-resolution (such as 320x240) images up through several mega-pixels images (such as 3000x3000 pixels). Significant performance improvements are realized if a model is developed that permits rendering of these huge images, but performed on low-resolution image data (such as 320x240). This provides not only quick display of images, but also reduces the amount of image data that must be transferred between data storage servers (where image data resides), image processing servers (where the image processing code renders the operations) and the client systems (where the output may be viewed).

20

This provides several benefits. First, the amount of processing power for each operation is significantly reduced since only a small amount of image data is actually processed. More importantly, in a network environment, only the low-resolution image data must be transmitted between server(s) and the remote client computer. This is particularly important when executing in a low-bandwidth (i.e. 56Kb modem) environment.

25

Another benefit realized by this invention is that when processing either resolution-independent or pseudo-resolution-independent image operations, the original image resolution does not have to be accessed, nor must it even be available. In some situations, the original image resolution is not available and may only be available when required for generation of high-resolution printed output. In this case, most of the processing must occur at a lower-resolution, either on the server computer on

30

WO 02/31754

PCT/US01/42694

the client / host computer. This invention provides a solution that results in consistent output even though the original image resolution may not be available. Without this, it would not be possible to guarantee consistent behavior except for resolution-independent image operations.

5

The processing of lower-resolutions images result in the need of less processing power and lower memory requirements, both of which are critical when developing low-cost consumer electronics digital image devices, including an information appliance, a digital camera, a digital camcorder, a digital television, a digital photo scanner, photo-enabled set-top box, a photo enabled game machine, a photo enabled internet device, cell phone, cable set-top box, WebTV™ or any other computing device that can view images.

Image operations that fall into the Resolution-independent category involve those that yield consistent results across all resolutions. For example, for image A, an imaging operation is applied to an image at a particular resolution and then the image is resized to a smaller resolution. For image B, the image is first resized to the smaller resolution and then the image operation is applied. If image A and image B are sufficiently visually close, taking into account the errors introduced during the resampling/resize operation, the operation is considered resolution-independent. Put another way, when the user views image A and image B side by side, they should visually appear the same.

Some of the following operations that fall into this category including rotation, cropping, translation, color adjustment, color twist, brightness/contrast adjustment (not based on an image's histogram), blur and sharpen operations (as defined by the FlashPix™ imaging model), Duotones (convert the image into a two tone color image), grayscale/black&white, negative, solarize, posterize (reduction in the number of colors), and bi-level/threshold. This is by no means a comprehensive list, but is illustrative of the type of operations that are considered resolution-independent. It should be noted, the FlashPix™ imaging model only supports a small subset of these operations.

WO 02/31754

PCT/US01/42694

There are several classes of pseudo-resolution-independent image processing operations. As additional resolution-dependent image operations are investigated to determine how they can simulate resolution-independent behavior, new classes of pseudo-resolution-independent operations will be created.

5 Histogram based Image Operations

This class of image operations modifies a given pixel value based a transform function from the histogram of the image. For example, an auto-enhancement function may look at the histogram and cut off 5% from each end of the histogram and redistribute the histogram across the entire range.

- 10 Since the histogram of the original resolution image is close to the histogram of a lower-resolution image this could possibly be considered resolution independent. However while this is generally correct, the assumption breaks down as the resolution becomes very small (such as a 192x192 thumbnail). Therefore, what is desired is an approach that provides true resolution-independence, but without
15 requiring processing of original resolution pixel data each time the image is rendered.

This invention solves this problem by performing the image operation in multiple phases. The Analyze phase is first performed on the image data. This phase
20 processes all the pixel data, ideally at the original image resolution, so that a histogram can be computed. During this phase, the pixels are not modified. Once the histogram is collected, appropriate image operation parameters can be ascertained, such as how the histogram should be redistributed. During the second phase, the Filter phase, the image data is actually modified. The benefit of this
25 approach is that the parameters are resolution-independent and thus, the Filter phase can actually be performed at any resolution, lower or higher than that of the original resolution of the image data.

- By performing an Analyze phase on the original image data's resolution and storing
30 those results for future renderings, the Filter phase can be performed at any resolution. This technique simulates resolution-independent behavior across all resolutions.

General Analyze phase / Filter phase utilization

WO 02/31754

PCT/US01/42694

The class of image operations described above is a subset of this category. While the Analyze/Filter phase model is described above in terms of usage for histogram filter operations, it can be used in a more general manner. For example, red-eye reduction and adjust artificial lighting are two image operations that are normally
5 considered resolution-dependent, but can simulate resolution-independence by allowing an Analyze phase to initially be performed at the original image resolution.

The Red-eye reduction filter operation is separable into two pieces: (a) detection of the red-eye and (b) removal / cleanup of the affected area. In order to accurately
10 detect the red-eye in the first place, it is desirable to perform the detection at the original image resolution. Otherwise, a subsampled image may sufficiently blur the area of the eye so that it cannot be accurately detected. Further, if processing occurred on a very low-resolution image, such as a 320x240 screen nail image, it most likely would not be possible to even visually see the red-eye.

15 Once the detection is performed at the original image resolution, the coordinate and radius of the red-eye is preserved in resolution-independent coordinates. With both the coordinate and radius, it is then possible to remove / cleanup the red-eye at any resolution. If cleanup were applied at a very low-resolution, such as a 320x240
20 screen nail, the cleanup region would be small. On the other hand, if performed on a very higher-resolution image, the cleanup region would be larger, centered about the coordinate of the red-eye.

The artificial lighting adjustment filter is also an operation that requires this two-
25 phased approach. The first pass is performed to determine how much of a particular light (such as yellow for incandescent lighting) exists in a photo. If more than a certain percentage (i.e. 25% of the photo) contains the particular color range, the image is corrected. Since the determination is based on this fixed percentage, the exact amount may vary slightly depending on the resolution. For example, if exactly
30 25% of the image contains yellow at the original image resolution. When the image is downsampled, the percentage calculated will usually be slightly below or above 25%.

WO 02/31754

PCT/US01/42694

For consistency across all resolutions, the detection of the color is performed at the original image resolution and stored during the Analyze phase. When rendering occurs, the Filter phase is performed at any resolution, but based on the data collected from the original image resolution data during the Analyze phase.

5 By performing an Analyze phase on the original image data's resolution and storing those results for future renderings, the Filter phase can be performed at any resolution. This technique simulates resolution-independent behavior across all resolutions.

Modification of internal parameters

10 The third class of pseudo-resolution-independent image operations modify internal parameter values, usually based on some constant value or range of values that becomes a function of the resolution of the image.

For example, the ripple filter simulates the effect of dropping a pebble into the pond rendered onto a photo. Most algorithms perform this by simulating a mathematical wave function across the image. In general though, this is dependent upon the size of the image. Therefore, a different number of ripples are seen on different resolutions of the same image.

20 An alternative approach is having the mathematical wave function that takes as a parameter the resolution of the image such that different resolutions will look similar. (i.e. have the same number of ripples) This is sometimes possible by scaling a parameter, based on the resolution, either linearly or by determining some logarithmic or geometric function to perform the scale.

25 Multiple discrete resolutions

The fourth class of pseudo-resolution-independent image operations selects different discrete internal parameter values for different resolutions. For these image operations, it is generally not possible to define a mathematical function for each internal parameter.

30 It is possible to simulate resolution-independent behavior by choosing discrete parameter settings for images at different resolutions, so that it is approximated across all resolutions. First, the maximum side of an image is determined. Next, the

WO 02/31754

PCT/US01/42694

maximum side is compared to each of the available resolutions. The one that most closely matches is chosen.

Paint Effects are a set of image operations that simulate various artistic effects, such as Chalk/Charcoal or Fresco art effect. For these operations, supported sizes might include: 128, 256, 512, 1024, 2048, and 4086. For images less than 128, the settings associated with 128 are used. There is nothing special about how these numbers are selected. Another image operation may choose to support the following sizes: 200,400,800,1600, and 3200. For any given image operation, a specific set of fixed resolutions is supported.

When an image is rendered that does not have an exact match to the supported size, two approaches are currently used. The most common approach is to choose the parameter settings closest to size of the image being rendered. This approach provides consistent results across all resolutions for many image operations in this class.

An alternative approach is to resize the image to match the correct intermediate size supported by the image operation. In some situations, this results in more consistent rendering across different resolutions, assuming the exact supported resolution of the image operation does not match the desired output resolution. After the operation is applied, the image may need to be resized back to the desired output resolution. If some or all of the image operations support the same size, this may be the more desirable approach since only one resample of the image is needed. Unfortunately, this does produce the negative effect introduced by the additional resampling operation.

Resolution-Dependent Operations

Image operations that fall into this category are by definition *not* resolution-independent. More importantly, these operations cannot approximate resolution-independence as described in the pseudo-resolution-independent section above.

WO 02/31754

PCT/US01/42694

The goal is to minimize the number of resolution-dependent image operations that are supported, or alternatively not support any resolution-dependent operation. In the past this was not viable since it would result in an application supporting very few imaging operations. By being able to simulate resolution-independence (i.e. psuedo-
5 resolution-independence) at continuous or discrete sets of resolutions, it is now feasible for an application to simply not support resolution-dependent operations as a viable option.

Rendering Implementation Details

10 In the best mode the rendering pipeline takes into account the characteristics of each of the above image operations, in order to determine how to render each of the image operations in the most efficient manner. These characteristics determine if the image operation is resolution independent, pseudo-resolution-independent, or
15 neither (i.e. resolution-dependent). Further, the relationship between various image operations must be defined since each operation may interact with the others. For example, the application of a Chalk/Charcoal paint effect followed by a rotate result in a different outcome compared to if the rotation were applied prior to the application of the paint effect.

20 If all operations are resolution-independent, the rendering pipeline is very similar to what is defined by the FlashPix™ image model. In this case, the relationship between the various image operations must still be specified, but all operations can occur on the same resolution. More importantly, all can be performed on low-
25 resolution image data thus requiring less processing power and memory.

Where the benefit of this invention becomes more apparent is when one or more of the image processing operations includes a pseudo-resolution-independent image
30 operation. In this case, rendering may be required at some predetermined resolution, close to the desired output resolution, but does not require the image data at the original resolution to be accessed or rendered.

Base Pipeline Implementation for this Invention

WO 02/31754

PCT/US01/42694

Now referring to Figure 4, a flowchart is shown detailing the process for rendering the image as defined by this invention. The process 400 begins at 402 when a particular digital image is opened by the application. At 404, the resolution of the digital image is determined and recorded in *originalImageResolution*. At 406, the
5 resolution of the output device is determined and the variable *finalOutputResolution* is set to this value.

At 408, it is determined, via process 500, what the initial working resolution should be and *currentOutputResolution* is set to this value. At 410, the image is copied to
10 an appropriate buffer and is resized to match the resolution specified in 408 and control is passed to 412.

At 412, a determination is made if any image editing operations should be performed. If no image operations are to be performed, control is pass to 450.
15 Otherwise, at 414 it is determined if the image operation is a resolution-independent operation. If it is not a resolution-independent operation, control is passed to 420 (figure 4a). Otherwise, control is passed to 416 where the image operation is performed at the *currentOutputResolution*. Control is then passed back to 412.

At 420 (figure 4a), it is determined if the operation is pseudo-resolution-independent
20 If it is, control is passed to 422. Otherwise, control is passed to 440 since it is a resolution-dependent operation. At 440, the *bestResolutionForImageOperation* is set to *originalImageResolution*, since it is a resolution-dependent operation. Control is then passed to 442

25 At 422, it is a pseudo-resolution-independent operation. There it is determined if a DoAnalyze phase is required. If it is required, then at 424 the DoAnalyze phase is handled, via process 600 after which control is passed to 426. If no DoAnalyze phase is required, control is passed directly to 426 where it is determined, via
30 process 700, what the closest supported resolution for this pseudo-resolution-independent image operation is and sets *bestResolutionForImageOperation* to this value. Control is then passed to 442.

WO 02/31754

PCT/US01/42694

At 442, if the *bestResolutionForImageOperation* matches *currentOutputResolution*, control is then returned to 416 (figure 4) where the image operation is performed on the image. If the resolutions do not match, control is passed to 444 where the image is resized to match the *bestResolutionForImageOperation*. At 446, the
5 *currentOutputResolution* is set to the *bestResolutionForImageOperation*. Control is then returned to 416 (figure 4) where the image operation is performed on the image.

At 450 (figure 4), all image operations have been performed and the values
10 *currentOutputResolution* and *finalOutputResolution* are compared. If these are equal, control is passed to 454 where the processed image is sent to the output display device. Otherwise, control is passed to 452 where the image is resized/resampled to match the *finalOutputResolution*. Control is then passed to 454 where the processed image is sent to the output display device.

15 It is clear from the flowchart that if all image operations are resolution-independent, and can be performed at any resolution, then all operations are performed at the *finalOutputResolution*. However, when an image operation is resolution-dependent, it must be performed at the original resolution of the image
(*originalImageResolution*). After all image operations have been performed, the
20 image must then be rescaled/resampled back the *finalOutputResolution* before it is passed to the output device.

In general, when processing the pseudo-resolution-independent operations, either
25 no resampling is required (*bestResolutionForImageOperation* and *finalOutputResolution* are the same), or a resolution much closer to the *finalOutputResolution* is used compared with requiring the *originalImageResolution* if it were a resolution-dependent operation.

Determination of the initial working resolution (Process 500 / Figure 5)
30 Several different heuristics are applied to determine the ideal resolution for a given image operation (*bestResolutionForImageOperation*). This value is dependent upon the image operation in question so a general approach is listed below.

WO 02/31754

PCT/US01/42694

A general rule is that it is best not to alter the *currentOutputResolution* if possible. This preserves image fidelity. Ideally, *bestResolutionForImageOperation* is the same at the *finalOutputResolution* such that resampling/rescaling as performed in step 452 of figure 4 is not needed. However, this sometimes is not possible.

5

The next approach is to find a resolution that is consistent and works across all image operations. In this way, at most two resample operations need to be performed: a resample from the *originalImageResolution* to the *targetResolution* (for this case, it is the same for all operations) and a final resample to the *finalOutputResolution* required by the output device. If all image operations can work at the same resolution, no additional resampling is needed. Process 500 and 700 take these rules and approaches into account during its computation.

It is a design goal of the pseudo-resolution-independent image operations to support resolution-independence for a specific discrete set of resolutions provided the continuous range model is not feasible. In the preferred embodiment, this discrete set is the same for all pseudo-resolution-independent operations. In this way, it reduces the amount of resampling required. Even if this is not possible, the pseudo-resolution-independent image operations are still preferred over resolution-dependent operations since rendering usually does not need to occur at the *originalImageResolution*.

Now referring to Figure 5, process 500 is shown that details the process of selecting the single best resolution for the series of image operations that are to be processed, based on *originalImageResolution* and the *finalOutputResolution*. Technically, this process is optional. One implementation of process 500 would be to simply return *finalOutputResolution* as "single best resolution". This is valid since process 700 results in the resize of the image to meet the needs of the specific image operation(s).

25
30

However, it is advantageous to limit the amount of resampling that occurs, since it results in image degradation. A more desirable approach is to opportunistically find the single best image size for all image operations that are being performed. This

WO 02/31754

PCT/US01/42694

may not always be possible, but in the preferred embodiment an attempt is made to find the best initial resolution.

The process 500 begins at 502 where *targetResolution* is set to the smaller of
5 *originalImageResolution* and *finalOutputResolution*. The assumption is that when
performing operations at less than the *originalImageResolution*, it is ok to perform
the operations at the *finalOutputResolution*. This is because the operations are
assumed to be resolution-independent or pseudo-resolution independent and result
in consistent output, even though it is processed at *finalOutputResolution*, opposed
10 to the *originalImageResolution*. However, if the operation is performed at greater
than the *originalImageResolution*, it is acceptable to perform it on the
originalImageResolution. This is a performance / memory tradeoff since it is
assumed that for this case, performing the operations at the *originalImageResolution*
and then resampling the image to match the *finalOutputResolution* is acceptable.
15 For some systems and operations this may not be desirable and in those
embodiments, the *targetResolution* is set to *finalOutputResolution* at 502 (see
section 4.1 on High Resolution Processing).

At 504, a determination is made if any image editing operations should be
20 performed. If no image operations are to be performed, then control is passed to
550. Otherwise, at 506 it is determined if all image editing operations are of type
resolution-independent. If true, then control is pass to 550. Otherwise, control is
passed to 510.

25 At 510, it is determined if all image operations are of type pseudo-resolution-
independent or resolution-independent. If false, control is passed to 530.
Otherwise, at 512, it is determined if all operations support the *targetResolution*. If
true, control is pass to 550. Otherwise, control is passed to 516.

30 At 516, it is determined if all the pseudo-resolution-independent image operations
support some "common resolution" closest to the *targetResolution*. By examining
each operation, it is determined if one resolution, which is closest to the
targetResolution, is supported. If this is true, then control is pass to 518 where the

WO 02/31754

PCT/US01/42694

targetResolution is set to this "common resolution" and control is passed to 550. Otherwise, control is passed to 530.

At 530, either no common resolution has been determined or the operation is resolution-dependent. Therefore, the *targetResolution* is set to
5 *originalImageResolution* so processing can be performed on the original image resolution, thus guaranteeing consistent results across all resolutions. Control is then passed to 550.

10 In the preferred embodiment, step 530 does not occur since some common *targetResolution*, close to the *finalOutputResolution*, is always determined for all supported imaging operations in the PictureIQ imaging system. If this were not the case, the pseudo-resolution-independent operation would be treated the same as resolution-dependent operations. In the future, additional work will be done, such as
15 reordering of operations so that those operations that are resolution-independent or are pseudo-resolution-independent and with a resolution close to the *finalOutputResolution* will be performed first.

At 550, the *targetResolution* is returned to the caller.

20

Handle DoAnalyze Phase (Process 600 / Figure 6)

Now referring to Figure 6, a flowchart is shown detailing the process for handling the DoAnalyze phase as defined by this invention. The process 600 begins at 602 where it is determined if results from a previous DoAnalyze phase exists and are still
25 valid. If this is true, control is passed to 650. Otherwise, at 604, it is determined if the DoAnalyze can be performed at the *currentOutputResolution*. If this is true, at 606 the DoAnalyze is performed at the *currentOutputResolution* and control is passed to 650. If this is not the case, the DoAnalyze must be performed at the *originalImageResolution* at 608.

30

At 608, it is determined if it is sufficient to resize the accumulated results at the *currentOutputResolution* to the *originalImageResolution*, or if all image operations must be reapplied to the original image at the *originalImageResolution*. If

WO 02/31754

PCT/US01/42694

reprocessing of all operations at the *originalImageResolution* is required, control is passed to 620. Otherwise, at 610 the working image is copied into a separate buffer. At 612, the working image is resized to the *originalImageResolution*. Control is then passed to 606, where the DoAnalyze is performed.

5

At 620, the original image at the *originalImageResolution* is retrieved. At 622, it is determined, based on the current image operation needing the DoAnalyze data, which operations in the sequence up to this operation are to be applied at the original resolution. This may vary depending on the operation. After the original resolution image has been processed, control is then passed to 606, where the DoAnalyze is performed.

10

At 650, the DoAnalyze phase is complete and all internal variables are persisted / saved. Control is then returned to the caller.

15

Determine closest supported resolution (Process 700 / Figure 7)

Now referring to Figure 7, process 700 is shown detailing the process of selecting the best resolution (*bestResolutionForImageOperation*) for a given image operations that is being processed, based on *originalImageResolution* and the *currentOutputResolution*. Process 700 starts at 702 where the closest supported resolution for the current image operation compared to the *currentOutputResolution* is determined and *bestResolutionForImageOperation* is set to this value.

20

Unable to find a consistent resolution for all operations.

25

In the preferred embodiment, it is always possible to find a close resolution, generally much smaller than the *originalImageResolution* that is supported by all pseudo-resolution-dependent operations. By definition, all PictureIQ image operations fall into this category.

30

In some embodiments, this might not be possible. When this occurs, the *originalImageResolution* can be used. However, it is generally not desirable to resample the image each time an operation is applied since this results in image degradation.

WO 02/31754

PCT/US01/42694

In another embodiment, if there are two or three common required resolutions needed for all image operations, it may be desirable to support each, with the overhead of additional resample operations to scale between the various operations
5 as needed. This is an application specific tradeoff that must be made based on the ability to re-order the operations.

Handling of Resolution-Dependent Image Operations

10 While it is true that the maximum benefit of this invention is realized when the image operations utilized are either resolution-independent or pseudo-resolution-independent, other benefits are still possible. For example, it may be possible to re-order the image operations such that the resolution-dependent operations can be performed later (or ideally last) in the sequence of all image operations that are to be performed.

15 By doing this, many of the image operations that are either resolution-independent or pseudo-resolution-independent can be performed at a lower resolution, ideally at or close to the *finalOutputResolution*. After those operations have been performed quickly at the *finalOutputResolution*, the image must be resampled to the
20 *originalImageResolution* for processing of the resolution-dependent operations. Finally, the image must then be resampled back to the *finalOutputResolution*. It is application dependent if the potential quality loss or the extra time/resources needed for the resample operations incurred is worth the tradeoff.

25 While the most significant benefit of this invention is the ability to perform an image operation on much lower resolution image data, other benefits are realized when processing higher resolution image data compared to the original.

30 Generally, most applications apply the image operations at the original image resolution and resize the image accordingly, smaller or larger, to match the output device. When outputting to a high-resolution print device, the resolution of the printer is generally much higher than that of the original image resolution. In this

WO 02/31754

PCT/US01/42694

case, the original image with all the operations applied is resized/resampled to a larger resolution to match the printer.

5 The drawback with this approach is that resampling to a larger resolution introduces other artifacts, including pixelation (blockiness) and other aliasing effects. This is because higher resolution image data must be "created" since it does not already exist. To circumvent this problem, high frequencies of the image (where pixelation is most prevalent) is slightly blurred to lessen the effects of the aliasing.

10 If an image processing operation is performed that attenuates the edges (such as a sharpen operation or accented edges), the resample operation may have the effect of either introducing blockiness or a degree of blurring. Each of these is not desirable.

15 An alternative approach, made possible by this invention, is to first resample the image to match the resolution of the output device, without apply any image operations. Next the various resolution-independent or pseudo-resolution-independent operations are performed at the high-resolution (presumably higher than the original image resolution). This is not possible for any operation that is
20 resolution-dependent since it would result in inconsistent results.

In general, care must be taken if this approach is used. This increases the processing time, and memory requirements, of each image processing operation since they are performed on very high-resolution image data. Further, if any
25 operations are resolution-dependent, they must be performed at the original image resolution

The more likely scenario is to use a hybrid approach that results in processing some of image operations at the original image resolution including all resolution-
30 dependent operations. Then the image is resampled and the subset of resolution-independent or pseudo-resolution-independent operations that exhibit undesirable artifacts as a result of the resample/resize operation is performed at the higher resolution. This assumes that some operations can be reordered.

WO 02/31754

PCT/US01/42694

This invention has been designed to provide consistent results across all resolution. As stated earlier, for image A, an imaging operation is applied to an image at a particular resolution and then the image is resized to a smaller resolution. For image
5 B, the image is first resized to the smaller resolution and then the image operation is applied. If image A and image B are sufficiently close, taking into account the errors introduced during the resampling/resize operation, the operation is considered resolution-independent. Put another way, when the user views image A and image B side by side, they should visually appear the same.

10

While in theory, if this assumption holds, the effects of resolution-independence are achieved. For some limited set of resolution-independent or pseudo-resolution-independent operations, this may not be entirely correct. Certain artistic effects exhibit this characteristic. When the effect is applied to the original resolution image,
15 it is visible. However, when the original resolution image, with the effect already applied, is resampled to a screen nail or thumbnail, it becomes less visible or not visible at all.

20

From an image-processing standpoint, the effect of resampling the image down to a smaller resolution reduces the effects of the image processing operation. For example, if a sharpen operation is applied at the original resolution and then the result is resampled to a smaller resolution, the amount of sharpen appears less at lower resolutions. This is because the resample operation tends to blur sharp edges and reduces the effect of the sharpen operation.

25

This section presents output that shows the results from using resolution-independent techniques when performing the same operation on several different resolutions, both from resolution-dependent and pseudo-resolution-independent operations. Figure 8 shows the original image at different resolutions.

30

Each section shows four resolutions: 100x67, 200x134, 400x286, and 800x536. The operations are applied at each resolution. Note, the 800x536 image has the operation applied at that resolution and is subsequently resampled down to fit into

WO 02/31754

PCT/US01/42694

this document. This allows for direct comparison between the 800x536 and 400x286 resolution. In theory, a pure resolution-independent operation should yield identical results when output is compared side-by-side at the same view size.

5 Figures 9 and 10 show the output for the ripple filter for a pseudo-resolution-independent version and a resolution-dependent version of the same filter. Notice how the appearance of the ripple effect changes with the resolution in figures shown for the resolution-dependent output. This is an example of a filter in which internal parameters to the algorithm are modified based on the resolution.

10

Figures 11 and 12 show the output for the mosaic tiles filter for both versions. Notice how the number of tiles changes based on the resolution in the figures shown for the resolution-dependent output. This is an example in which the internal parameters to the algorithm are modified based on the resolution.

15

Figures 13 and 14 show the output for the Fresco paint effect. Notice the significant difference between the various resolutions for the resolution-dependent output. This is an example in which a discrete set of resolutions is supported. To the keen eye, some differences may be seen between different resolutions of the figures shown for the resolution-independent output, but still very similar. Clearly, this is much better than the resolution-dependent output.

20

It is important to note, if an application using today's technology (without this invention) wants to achieve the same results across all resolutions, the processing must be performed at the original image resolution. For the Fresco effect, the original image at 1600x1072 takes 11 seconds to complete processing on that resolution running on a 750 MHz Intel Pentium III, in comparison to less than 1 second for the same rendering of a 400x286 or smaller image using pseudo-resolution-independent techniques.

25

Figure 15 shows side-by-side results from the output for the Chalk/Charcoal paint effect using pseudo-resolution-independent and resolution-independent techniques. This is an example in which a discrete set of resolutions is supported. For

30

WO 02/31754

PCT/US01/42694

comparisons, it is sometimes helpful to view the same sized output side-by-side since these techniques try to approximate the same results across all resolutions. The best way for the comparison, at the pixel level, is to view each image at the same size, regardless of resolution.

5

The comparison shows how the different resolutions compare when viewed at the same size. In this case, the 200x134 is viewed at 100%, but the 400x268 is viewed at 50% and the 800x536 is viewed at 25%. When viewed this way, a more accurate determination can be made about how close the resolutions compare side-by-side.

10

For Chalk/Charcoal, more differences are visible between resolutions using the pseudo-resolution-independent technique. The primary reason for this is due to the fact that when working at lower-resolutions, a certain amount of data is already lost and must be approximated. The same situation exists for a high-resolution photo that is resampled down to a screen nail (320x240) and then enlarged (zoomed in) for comparison with the original. Clearly, there is data loss similar to what is shown in the figures in section 5.8. This should be expected. It is just that certain image operations can perform better at lower-resolutions and are able to "better create" this resolution.

15

20

While the present invention has been described as being used with a digital image system (video or still), it should be appreciated that the present invention may generally be implemented on any suitable digital image system. This includes a PC-based imaging application, Web sharing applications that permit sharing, distribution, or viewing of image data between a central server and a client, as well as direct end-user peer-to-peer connected systems.

25

It can be included as part of an embedded information appliance or digital image device and can work equally well in both a *wired* network environment as well as a *wireless* environment.

30

This disclosure of a system and method to render image data to simulate pseudo-resolution-independent behavior according to the preferred embodiments of the present invention is merely exemplary in nature and is no way intended to limit the

WO 02/31754

PCT/US01/42694

invention or its application or uses. Further, in the above description, numerous specific details for implementation are set forth to provide a more thorough understanding of the present invention disclosure. It will be apparent, however, to one skilled in the art, that the present invention may be practiced without these
5 specific details. In other instances, characteristics and functions of the well known processes have not been described so as not to obscure the present invention.

WO 02/31754

PCT/US01/42694

Claims:

We claim:

- 5 1. A method for applying normally resolution-dependent image effects to a digital image in such a manner as to enable a rendering of said digital image at any resulting resolution while substantially maintaining the appearance of the image effect, comprising the steps of:
- 10 a. determining what imaging effect parameters will substantially modify the imaging effect results when rendered at a different resolution; and
- b. modifying said parameters to account for the resultant rendering resolution.
- 15 2. The method of claim 1 wherein the imaging effect is based on the transform function derived from a histogram of the image and further including the steps of:
- a. analyzing pixel data of the image at an original resolution; and
- b. determining the appropriate image operation parameters; and
- c. modifying said parameters subordinate to a predetermined resulting resolution.
- 20 3. The method of claim 1 wherein the imaging effect is based upon a coordinate and radius of particular area of a digital image and further including the steps of:
- a. preserving coordinate and radius data in terms relative to the original resolution; and
- 25 b. performing the desired imaging effect at a resulting resolution with the radius and coordinate information modified by that resulting resolution.
4. The method of claim 3 wherein the imaging effect is red-eye reduction.
5. The method of claim 1 wherein the imaging effect is an artificial lighting adjustment further including:
- 30 a. a spectral analysis of quantity of each range of light frequency in the image and determining that quantity in terms of a percentage; and
- b. applying an imaging effect using that percentage; and

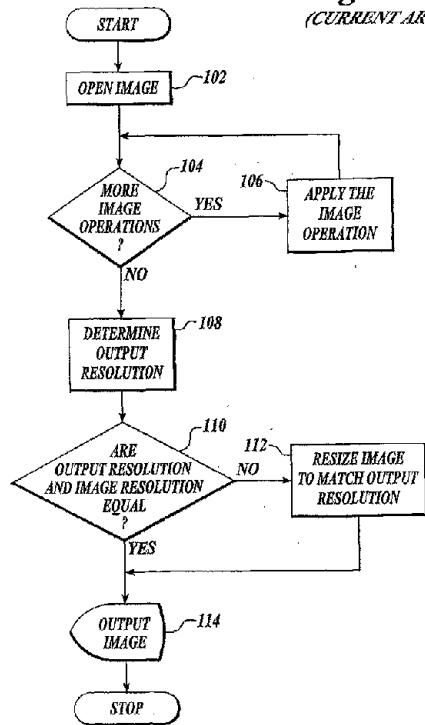
WO 02/31754

PCT/US01/42694

- c. upon rendering the image at any resolution, that determined percentage of light is kept substantially constant .
6. The method of claim 1 wherein the imaging effect is based on a value dependent upon determining an image resolution and further including the step of scaling that dependent value, based on the resulting resolution required.
7. The method of claim 1 wherein the imaging effect is based on different discrete internal parameter values, thereby requiring the selection of a subset of appropriate values dependent on the resulting resolution required.
8. A method for applying techniques to the resolution dependent characteristics of a digital image comprising the steps of:
- a. opening the image;
 - b. determining the image resolution;
 - c. determining an output device resolution;
 - d. selecting an optimal resolution upon which to apply an image effect; and
 - e. applying the image effect to the optimal resolution selected.

20

Fig. 1A.
(CURRENT ART)



2/17

Fig. 1B.
(CURRENT ART)

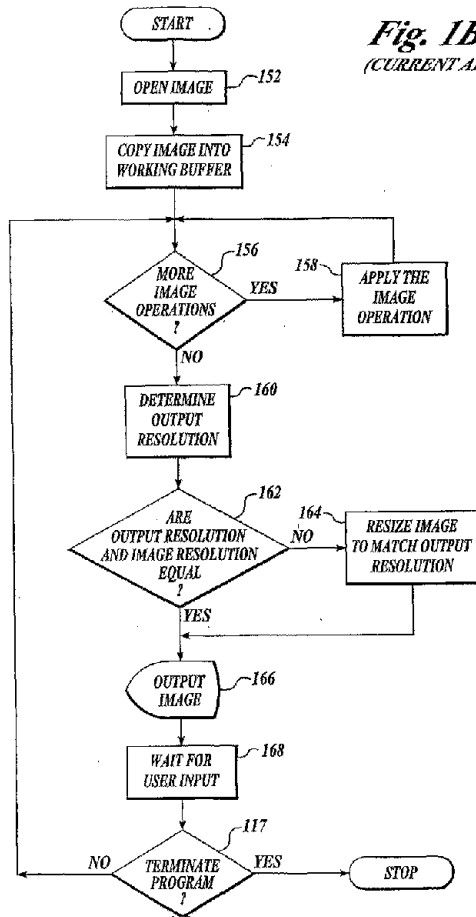
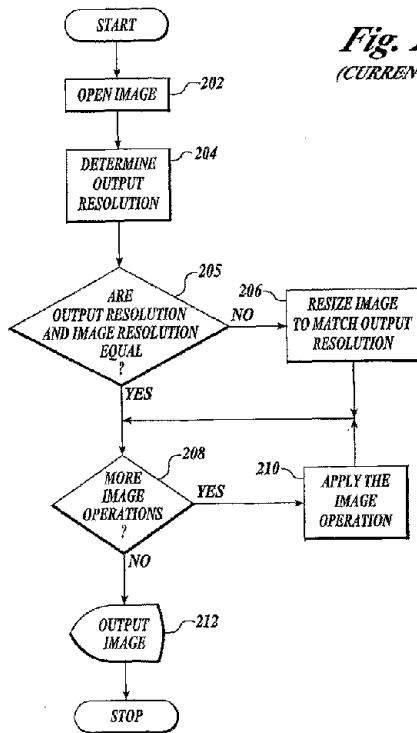


Fig. 2.
(CURRENT ART)



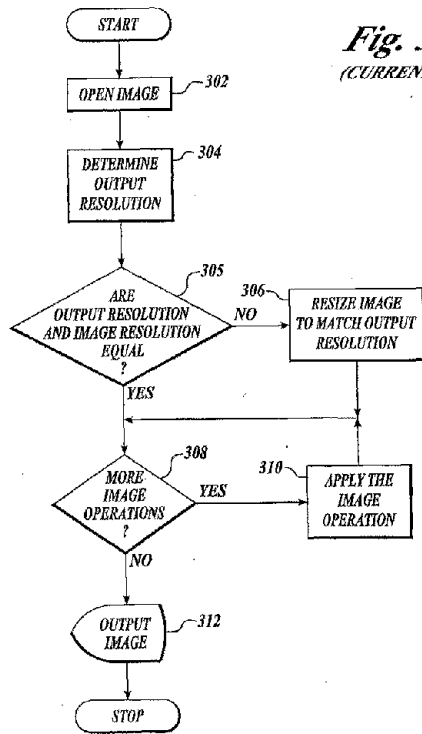


Fig. 3.
(CURRENT ART)

Fig. 4.

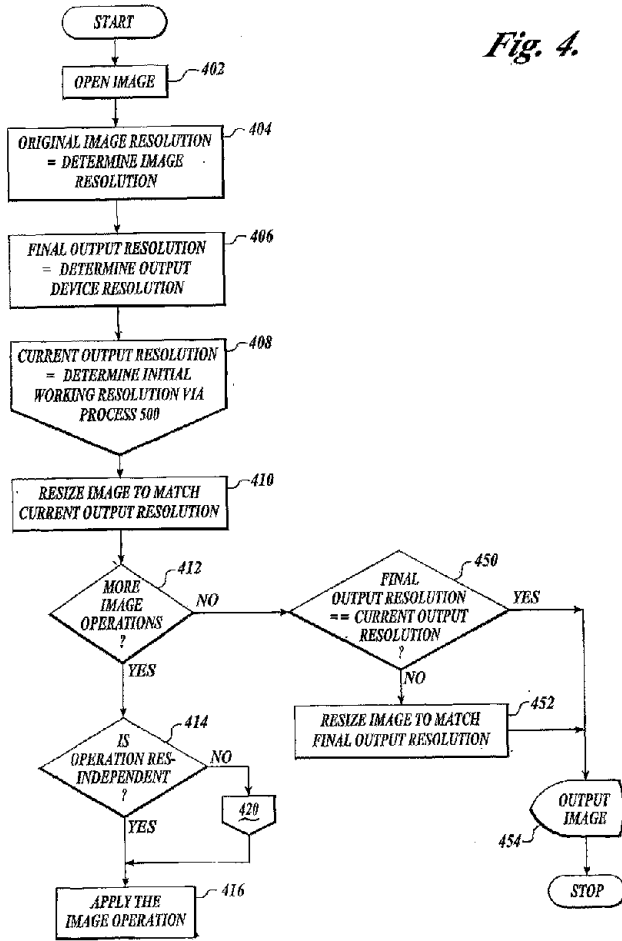
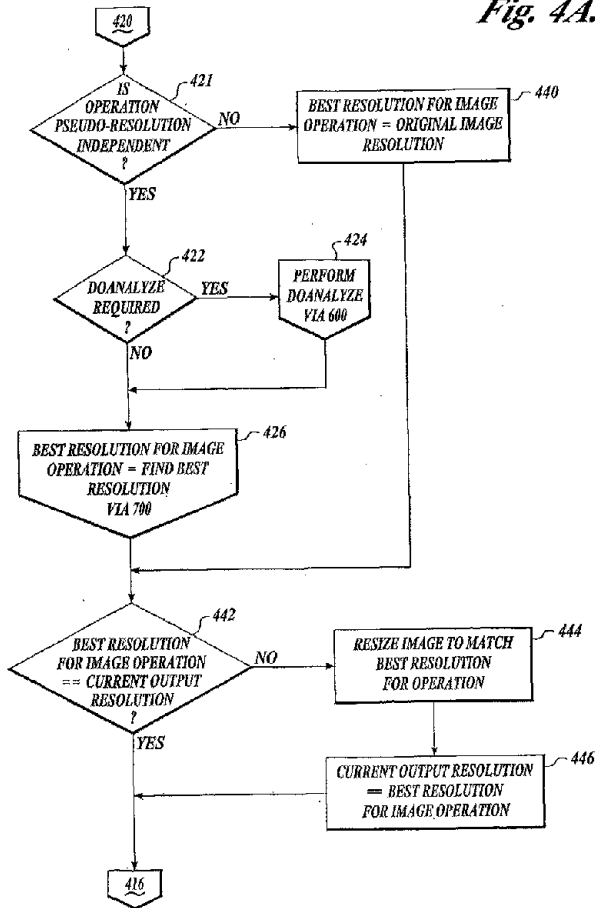
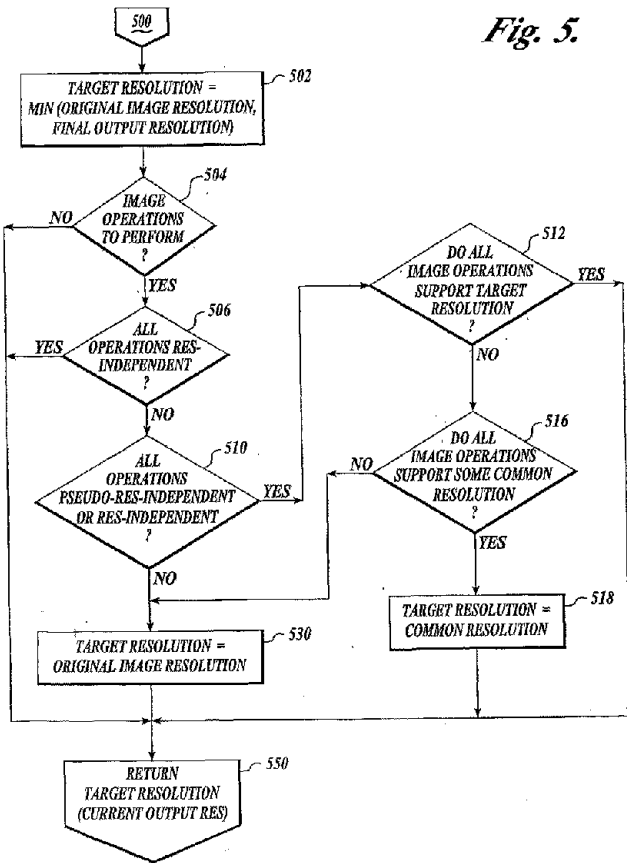


Fig. 4A.



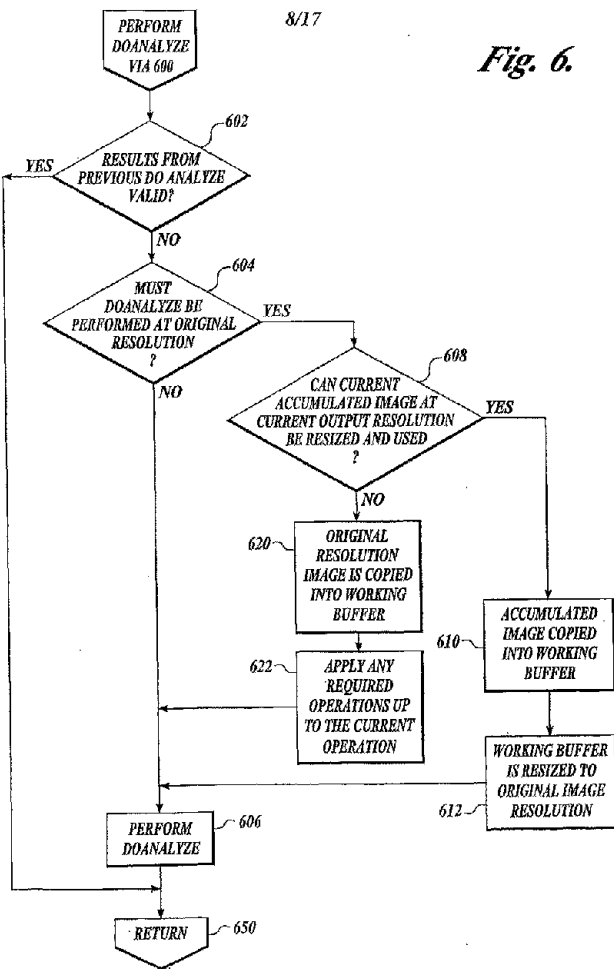
7/17

Fig. 5.



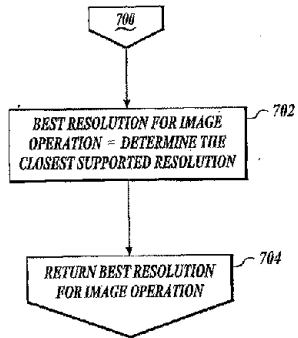
8/17

Fig. 6.



9/17

Fig. 7.



WO 02/31754

PCT/US01/42694

10/17



ORIGINAL: 100x67



ORIGINAL: 200x134



ORIGINAL: 400x268



ORIGINAL: 800x536 RESAMPLED DOWN TO 400x268

Fig. 8.

SUBSTITUTE SHEET (RULE 26)

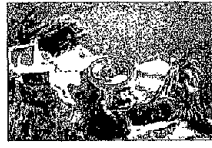
WO 02/31754

PCT/US01/42694

11/17



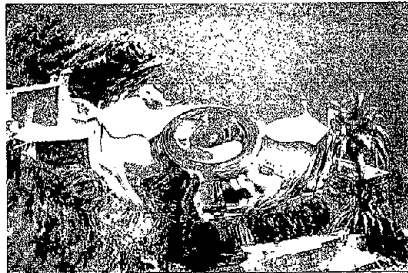
RESOLUTION-INDEPENDENT RIPPLE: 100x67



RESOLUTION-INDEPENDENT RIPPLE: 200x134



RESOLUTION-INDEPENDENT RIPPLE: 400x268



RESOLUTION-INDEPENDENT RIPPLE: APPLIED TO 800x536 AND RESAMPLED DOWN TO 400x268

Fig. 2
SUBSTITUTE SHEET (RULE 26)

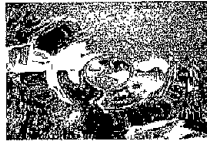
WO 02/31754

PCT/US01/42694

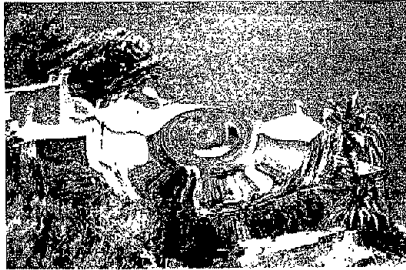
12/17



RESOLUTION-DEPENDENT RIPPLE: 100x67



RESOLUTION-DEPENDENT RIPPLE: 200x134



RESOLUTION-DEPENDENT RIPPLE: 400x268



RESOLUTION-DEPENDENT RIPPLE: APPLIED TO 800x536 AND RESAMPLED DOWN TO 400x268

Fig. 10

SUBSTITUTE SHEET (RULE 26)

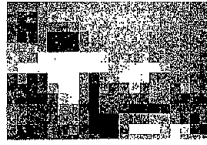
WO 02/31754

PCT/US01/42694

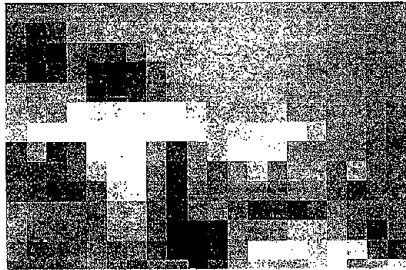
13/17



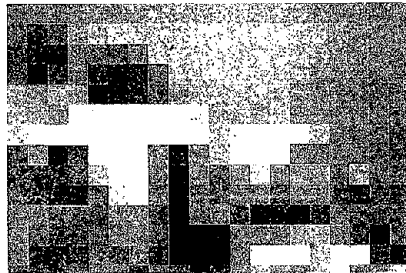
RES-INDEPENDENT MOSAIC TILES: 100x67



RES-INDEPENDENT MOSAIC TILES: 200x134



RESOLUTION-INDEPENDENT MOSAIC TILES: 400x268



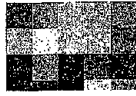
RESOLUTION-INDEPENDENT MOSAIC TILES: APPLIED TO 800x536 AND RESAMPLED DOWN TO 400x268

Fig. 11
SUBSTITUTE SHEET (RULE 26)

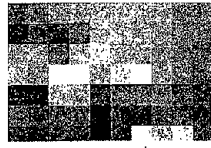
WO 02/31754

PCT/US01/42694

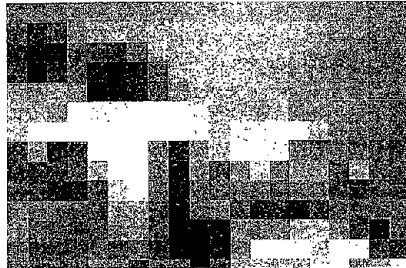
14/17



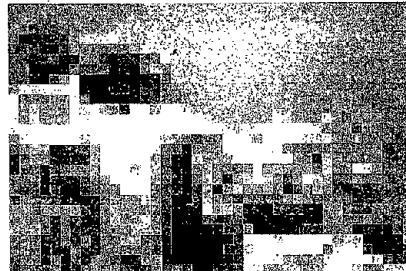
RES-DEPENDENT MOSAIC TILES: 100x67



RES-DEPENDENT MOSAIC TILES: 200x134



RESOLUTION-DEPENDENT MOSAIC TILES: 400x268



RESOLUTION-DEPENDENT MOSAIC TILES: APPLIED TO 800x536 AND RESAMPLED DOWN TO 400x268

Fig. 12.
SUBSTITUTE SHEET (RULE 26)

WO 02/31754

PCT/US01/42694

15/17



RESOLUTION-INDEPENDENT FRESCO: 100x67



RESOLUTION-INDEPENDENT FRESCO: 200x134



RESOLUTION-INDEPENDENT FRESCO: 400x268



RESOLUTION-INDEPENDENT FRESCO: APPLIED TO 800x536 AND RESAMPLED DOWN TO 400x268

Fig. 13

SUBSTITUTE SHEET (RULE 26)

WO 02/31754

PCT/US01/42694

16/17



RESOLUTION-DEPENDENT FRESCO: 100x67



RESOLUTION-DEPENDENT FRESCO: 200x134



RESOLUTION-DEPENDENT FRESCO: 400x268



RESOLUTION-DEPENDENT FRESCO: APPLIED TO 800x536 AND RESAMPLED DOWN TO 400x268

Fig. 14.

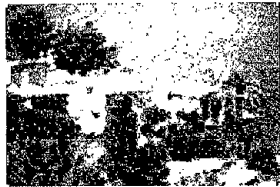
SUBSTITUTE SHEET (RULE 26)

RESOLUTION-INDEPENDENT APPROACH

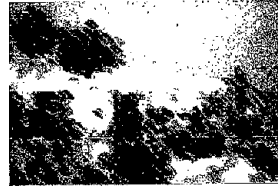
RESOLUTION-DEPENDENT APPROACH



APPLIED TO 100x67



APPLIED TO 200x134 IMAGE



APPLIED TO 400x268 IMAGE AND RESAMPLED DOWN TO 200x134



APPLIED TO 800x536 IMAGE AND RESAMPLED DOWN TO 200x134



Fig. 15.

【 国際調査報告 】

INTERNATIONAL SEARCH REPORT		International application No. PCT/US01/42592
A. CLASSIFICATION OF SUBJECT MATTER IPC(Cl) : G06K 6/02 US CL : 382/269 According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) U.S. : 368/149, 398-800; 368/12, 13 Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) WEST search terms: image, resolution, edit, effect, process		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5,805,721 A (VUYLSTEKEB et al) 08 September 1998, Abstract.	1-8
A	US 6,088,489 A (MIYAKE) 11 July 2000, Abstract.	1-8
A,P	US 6,298,151 B1 (JODOIN et al) 02 October 2001, col. 2, lines 3-8.	1-8
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.		
<small>* Special categories of cited documents: "A" Document defining the general state of the art which is not considered to be of particular relevance "E" Earlier document published on or after the international filing date "L" Document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" Document referring to an oral disclosure, use, exhibition or other means "P" Document published prior to the international filing date but later than the priority date claimed "PP" Late document published after the international filing date or priority date and in conflict with the application, but cited to understand the principle or theory underlying the invention "X" Document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" Document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "Z" Document member of the same patent family </small>		
Date of the actual completion of the international search 17 JANUARY 2002		Date of mailing of the international search report 12 FEB 2002
Name and mailing address of the ISA/DIS Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 501-5630		Authorized officer Stephen Brinich Telephone No. (703) 501-4250 <i>Rugenia Zagan</i>

フロントページの続き

(72)発明者 ウィルキンズ、デイビッド

アメリカ合衆国 0 2 9 0 6 ロードアイランド州 プロビデンス グレン ドライブ 1 5

Fターム(参考) 5B050 BA06 DA02 DA04 EA07 EA08 EA09 EA12 EA14 FA02 FA03

FA05

5B057 CD03 CD05 CE03 CE04 CE17 DC23