

携帯型 AR 迷路アプリケーションの開発

浅賀 一樹^{1,a)} 藤代 一成²

概要: 近年, コンピュータを用いて素材画像から迷路を自動生成するアルゴリズムに関する研究が多数行われている. 本稿では, 携帯端末のカメラ機能を介して入力した素材画像と生成した迷路を拡張現実 (Augmented Reality; AR) 表示するような新たな迷路生成アプリケーションを提案する. AR による迷路の重畳表示に加えて, パノラマ画像入力やクロスカーソルによる没入的なプレイ方法を採用入れることにより, 「現実空間」から「迷路空間」への変換のみに着目していた既存研究とは異なり, 「現実空間」と「迷路空間」を融合した「AR 迷路」という新たなスタイルのエンタテインメント創成の実現を目指す.

キーワード: 迷路生成, 拡張現実, 携帯端末, 同心状パノラマ, クロスカーソル, エンタテインメント

Development of a Portable AR Maze Application

KAZUKI ASAGA^{1,a)} ISSEI FUJISHIRO²

Abstract: In recent years, many algorithms have been proposed, which generate mazes automatically from an input image. In this article, we present a new portable maze application which utilizes the camera function of a mobile terminal to superimpose a generated maze onto the input image in real time. Our application not only pursues AR (augmented reality) display of the maze, but also takes full advantage of panorama image input and immersive play with a cross cursor. In this way, we intend to propose a new style entertainment called “AR maze”, which differs from existing research results where their attentions are paid only to the plausible conversion from “real space” to “maze space”.

Keywords: Maze generation, augmented reality, mobile device, concentric panorama, cross cursor, entertainment.

1. 序論

芸術・デザインの分野において長年受け継がれてきた作品として, 迷路・迷宮がある. 二次元の画像から迷路を自動生成するアルゴリズムに関するコンピュータグラフィックス (CG) の研究も現在までに多く行われてきた. しかし, その大部分の研究は実装媒体として PC を使用している. 近年, スマートフォンやタブレット端末が普及してきたことにより, それら携帯端末上で実装を行う研究が増加している. 現在の携帯端末にはタッチパネルや加速度セン

サなど様々な機能が搭載されており, それらの機能を端末の携帯性と組み合わせることにより, PC とは異なる操作を実現できるからである.

例えば迷路の自動生成研究において, カメラ機能と端末の携帯性を組み合わせると, ユーザは時間・場所を問わず迷路生成のもととなる入力画像を取得することが容易となる. また生成された迷路を端末画面上に表示する際, カメラを介して元の物体が映る位置に常に重ねて表示し続けることで, 現実世界と仮想世界を融合した拡張現実 (Augmented Reality; AR) の空間を作ることが可能となる. ユーザが端末のカメラの視線方向を変えると, それに応じてカメラを介して元の物体上に表示される迷路画像の位置も変化し, ユーザはあたかもその空間に迷路が実在しているかのような感覚を得ることが可能となる.

¹ 慶應義塾大学 大学院理工学研究科
Graduate School of Science and Technology, Keio University

² 慶應義塾大学 理工学部情報工学科
Department of Information and Computer Science, Faculty of Science and Technology, Keio University

a) asaga@fj.ics.keio.ac.jp

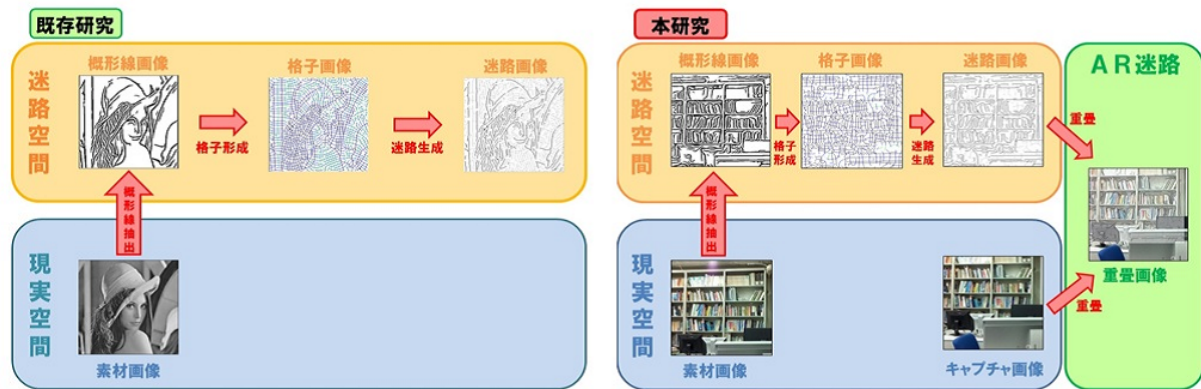


図 1 既存研究と本研究の比較.

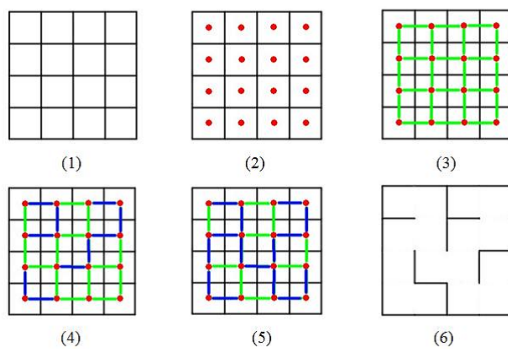


図 2 格子構造からの迷路生成の流れ [1]. 格子構造 (1) を無向グラフに変換し、最小極大木を生成した後に対応する壁を取り除いていくことで最終的な迷路が生成される (6) .

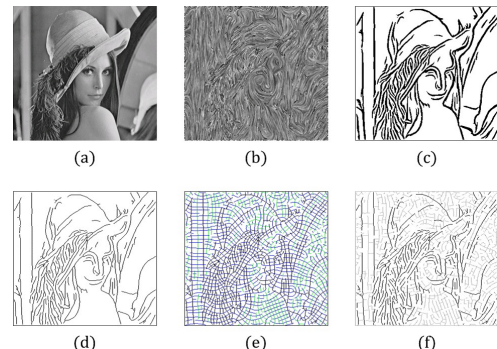


図 3 ETF を利用した迷路生成の流れ [2]. 入力画像 (a) から ETF を用いて概形線を抽出することで格子を形成し、最終的な迷路が生成される (f) .

本研究では、このように携帯端末のカメラ機能を介して入力した素材画像と生成した迷路を AR 表示することにより、PC における実装では実現しづらい画像入力の「手軽さ」やプレイ時における「没入感・臨場感」などの特徴をもった、新たな迷路生成アプリケーションを提案する。「現実空間」から「迷路空間」への変換のみに着目していた既存研究とは異なり、「現実空間」と「迷路空間」を融合した「AR 迷路」という新たなスタイルのエンタテインメント創成の実現を目指している (図 1) . また、AR 技術を利用した迷路画像の重畳表示に加えて、同心状パノラマ画像を入力に用いたり、クロスカーソルによる没入的なプレイ方法を採り入れることで、ユーザに対しより効果的に「没入感・臨場感」を与えることが可能なアプリケーションとなっている。

本稿は次節以降、以下のように構成されている。まず次節で迷路生成の関連手法および拡張現実における特徴点抽出・特徴量検出の関連手法を挙げ、3 節で入力画像からの迷路生成、AR 技術を利用した迷路画像の表示、同心状パノラマ画像を利用した画像入力やクロスカーソルによるプレイといった本研究で開発するアプリケーションの特徴およびその実装結果を述べる。4 節で本研究の評価および考察を行い、5 節で本稿をまとめ、今後の課題に言及する。

2. 関連研究

関連研究として、2.1 項で格子構造を利用することで入力画像から自動的に迷路生成を行う関連手法について述べ、2.2 項で、AR 技術における特徴点抽出・特徴量検出手法、特に携帯端末上でも高速処理が可能となるよう改良された関連手法について述べる。

2.1 格子構造を利用した迷路生成手法

現在までに、入力画像から自動的に迷路を生成する手法は多数存在している。なかでも代表的なものとして、Shivers[1] の手法が挙げられる。まず、入力画像からセルとその壁で構成される格子構造を形成し、それらをそれぞれ無向グラフの頂点と辺に対応させる。その後最小極大木を生成する無向グラフのアルゴリズムを利用して辺の本数を減らし、元の格子構造から対応する壁を取り除いていくことで迷路が完成する (図 2) . また、迷路のもととなる格子を形成する際に、画像の抽象化手法を用いて入力画像から概形線を抽出し、格子構造の辺の一部とすることで、「元画像に似た構造をもつ迷路」を生成している。

Wong ら [2] はこの無向グラフのアルゴリズムを用いた迷路生成手法をもとに、Kang ら [3] が提案した、画像の

辺方向における特徴量を保存するベクトル場である Edge Tangent Flow (ETF) による概形線抽出法を、格子形成の際に利用した迷路生成手法を提案している (図 3)。また入力に 2 枚の画像を使用しており、それぞれが迷路の概形および解となるような高水準の迷路生成を実現している。

本研究では、Wong ら [2] の手法を利用し「元の画像に似た構造をもつ迷路」を生成することで、変換した「迷路空間」を元の「現実空間」とより融合しやすいものになっている。

2.2 特徴点抽出・特徴量検出における高速化手法

AR 技術における特徴点抽出・特徴量検出手法は SIFT [4] や SURF [5] など数多く存在するが、これらの手法は高精度である一方、携帯端末上での実装を考えると計算量が多く、実行速度という観点から見て適した手法ではない。そこで、より高速な特徴点抽出手法として、コーナーの判定に決定木の学習を採り入れた FAST [6]、より高速な特徴量検出手法として特徴量をバイナリコードで記述する BRIEF [7] がそれぞれ提案された。しかし、これらの手法には回転不変性をもたないという問題点があるため、Rublee ら [8] は、FAST および BRIEF に「特徴点の向き」という概念を導入して改良した手法である Oriented FAST and Rotated BRIEF (ORB) を提案した。

本研究では、ORB を特徴点抽出・特徴量検出手法として利用することで、カメラから得られたキャプチャ画像上の特徴点抽出・特徴量検出および素材画像上の特徴点との比較を行い、迷路画像の重畳表示を行う。これにより、携帯端末という限られた処理能力をもつ端末においても高速に処理を行うことを可能としている。

3. アプローチおよび実装結果

本研究のアプローチおよびその実装結果について述べる。実装端末として、タブレット端末 XOOM (2 コア CPU, 1.0GHz, Android OS 4.0.3) を使用する。

本研究で開発するアプリケーションでは、まず携帯端末のカメラ機能もしくは画像ギャラリー機能を利用して素材画像を取得し、ETF を用いたフィルタリング処理を施すことで素材画像から概形線を抽出する。

次に、ETF を流線により可視化したものと抽出した概形線を融合させることで格子構造を形成し、無向グラフへ変換する。その後、無向グラフのアルゴリズムを用いて対応する格子上の壁を一つずつ取り除き、迷路を生成する。

迷路画像が得られたら、ORB による特徴点抽出・特徴量検出を利用して携帯端末のカメラキャプチャ画像と迷路画像を融合し、AR 表示を行う。ユーザは、端末画面内に AR 表示された迷路上を、端末を実際に動かすことでなぞるようにスタートからゴールまでカーソルを導くといった形でプレイを行う。

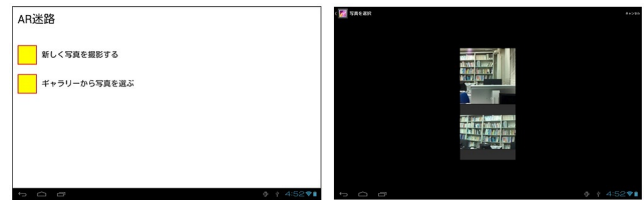


図 4 素材画像の入力画面。画像ギャラリー機能を利用して過去に撮影した画像を入力することも可能である。

3.1 素材画像の入力

本アプリケーションでは、携帯端末のカメラ機能を用いて写真を撮影することで、迷路のもととなる素材画像の入力を行う。また、携帯端末に搭載されている画像ギャラリー機能を利用することで、ユーザがアプリケーション実行以前に撮影した写真や、他端末で撮影した写真なども入力することが可能である (図 4)。

このように、素材画像の入力に携帯端末に搭載されているカメラ機能や画像ギャラリー機能を用いることで、ユーザにとって時間・場所問わず手軽に迷路生成が可能なアプリケーションとなっている。またこのとき、後の工程で利用するため入力画像から ORB を用いた特徴点抽出および特徴量検出も行っている。

3.2 迷路の生成

3.1 項で得られた入力画像に対し、Wong ら [2] の手法を利用することで、迷路のもととなる格子の形成および迷路の生成を行う。これにより、「元の画像に似た構造をもつ迷路」を生成することが可能となる。

3.2.1 概形線抽出を利用した格子形成

入力画像が得られたら、まず初めに画像内の各ピクセルにおいて周囲のピクセルとの輝度勾配から ETF を生成する。次に入力画像に対し、生成した ETF を利用して、Kang ら [3] が提案した色の平滑化フィルタである Flow-Based Bilateral (FBL) フィルタ、およびエッジ抽出フィルタである Flow-Based Difference-of-Gaussians (FDoG) フィルタという 2 種類のフィルタによる処理を行うことで、概形線を抽出する。

このように ETF を用いて概形線抽出を行うと、従来の画像抽象化手法よりも抽出される概形線が滑らかに続くため、少ない辺の本数でも、より迷路生成のもととなった画像の構造を把握しやすい迷路を生成することが可能となっている。また、ここで得られた概形線は、格子構造の壁として利用しやすくするための前処理として細線化処理が施される。

その後、ETF に平行な方向および垂直な方向の 2 方向において、ベクトル場を一定間隔の空いた複数の流線で可視化し、先程の細線化された概形線と融合させることで、概形線を壁の一部として利用した格子構造を形成する。

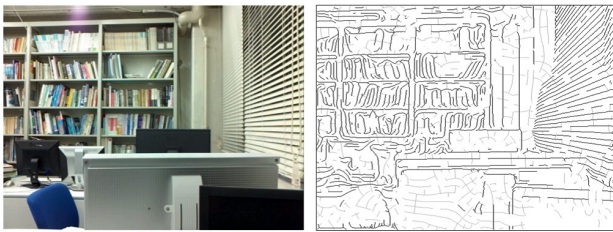


図 5 迷路生成の結果画像. 左: 入力画像, 右: 迷路画像.

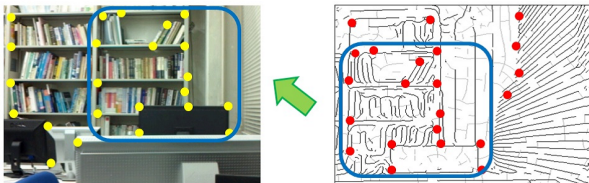


図 6 ORB を用いた画像のマッチング.

3.2.2 格子構造を利用した迷路生成

入力画像から格子構造が得られたら, 格子構造を無向グラフへ変換する. そして, Kruskal 法 [9] を用いて最小極大木を生成し, 対応する壁を一つずつ取り除いていくことで最終的な迷路が生成される. また, 全ての頂点がただ1つの同じ木に所属し, かつ環状構造をもたないという最小極大木の性質上, 最終的に生成される迷路は全てのセルが1つに繋がっており, どのセルからスタートしても迷路上の全ての場所に到達可能なものとなっている.

図 5 に, 入力画像から迷路生成を行った結果画像を示す. 図 5 左の画像 (大きさ 640 × 480 ピクセル) から右の画像を作成するのに約 117 秒を要した. また, 図 5 における迷路画像を AR 技術を用いて端末画面上に重畳表示させた結果, ORB による特徴点抽出・特徴量検出 (抽出した特徴点の個数 500 個) およびキャプチャ画像とのマッチング処理に, 1 フレームにつき約 1 秒を要した.

3.3 AR 技術を利用した迷路の表示

3.2 項で生成した迷路画像を, 携帯端末のカメラを介して元の物体が映る位置に透過させた状態で常に重畳表示し, 「現実空間」と「迷路空間」を融合させる.

まず, 1 フレーム毎にカメラに入力されるキャプチャ画像に対して, ORB を使用することでコーナーなど周辺との輝度変化が顕著な点である特徴点を抽出し, その輝度差を示した量である特徴量を検出する. その後, あらかじめ素材画像から抽出しておいた特徴点のもつ特徴量と比較することで, それぞれマッチング処理を行い, 迷路画像をキャプチャ画像の空間上に投影するための変換行列を計算する. そして最終的に, 迷路画像を透過処理したものを変換行列を用いて変形しキャプチャ画像上に投影することで, 現実空間の対応する位置に迷路画像が重畳表示される (図 6).

本研究では, オープンソースの画像処理ライブラリである OpenCV に実装されている ORB クラスを利用し特徴点



図 7 AR 技術を用いた画像の表示例.



図 8 回り込みによる視界の変化. 片方の視点から迷路を生成した場合, もう片方の視点では視界の変化が大きいため, 迷路の重畳表示が正確に行われなくなる.

抽出・特徴量検出を行っている. 図 7 に, 迷路画像とカメラからの入力画像を AR 技術を用いて融合・重畳表示させたキャプチャ画像を示す.

3.4 同心状パノラマ画像を利用した画像入力

AR 技術を利用した迷路の表示において, 図 7 に見られるように重畳表示されている部分とされていない部分の境界が見えてしまうことや, 図 8 に見られるように「撮影対象を中心にユーザ自身が回り込むように移動する」等の行為により視界が著しく変化し重畳表示がされなくなってしまうことは, ユーザがアプリケーションに対して感じる「没入感・臨場感」に大きく影響を与えてしまう原因となる.

本研究ではこれらの問題点を, ユーザ中心に撮影された同心状パノラマ画像を入力画像に利用し, 迷路生成する領域の拡大, すなわち最終的に重畳表示する領域の拡大を行い, さらに撮影時のユーザの位置を固定することにより暗黙的にプレイ時のユーザの位置を制限することでそれぞれ解決している.

パノラマ画像の入力は携帯端末に搭載されているカメラのオプション機能として実装されているパノラマ撮影機能を利用することで撮影を行い, ギャラリー機能から入力することで実現している. 図 9 に, 同心状パノラマ画像から迷路生成を行った結果画像を示す. 図 9 左の画像 (大きさ 1,280 × 547 ピクセル) から右の迷路画像を作成するのに約 284 秒を要した. また, 図 9 における迷路画像を AR 技術を用いて端末画面上に重畳表示させた結果, ORB による特徴点抽出・特徴量検出 (抽出した特徴点の個数 500 個) およびキャプチャ画像とのマッチング処理に, 1 フレームにつき約 1.5 秒を要した.

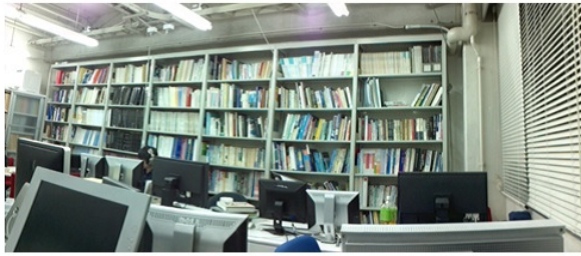


図 9 同心状パノラマ画像を用いた迷路生成. 左: 入力画像, 右: 迷路画像.

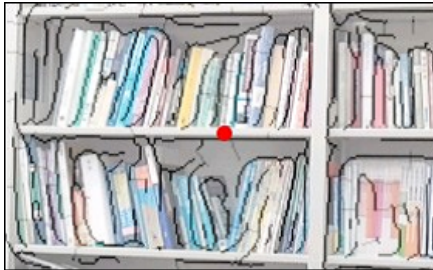


図 10 クロスカーソルを用いたプレイ画面の拡大図.



図 11 完成した迷路画像. 青い点がスタート地点, 緑の点がゴール地点.

3.5 ユーザによるプレイ

カメラによる入力画像から迷路画像が生成された後, ユーザが迷路上の好きな位置にスタートとゴールを指定することで実際にプレイ可能な迷路が完成する. 3.2 項で述べたように, 迷路の性質上ユーザがどこにスタートとゴールを設定してもクリアすることが可能となっている.

迷路のプレイ時は, AR 技術により迷路が重畳表示された端末画面上の中心部分にそれ自身は動かない丸いカーソルを表示する. その後, ユーザによる端末の動きに合わせて重畳画像側を動かすことで, カーソルをスタートからゴールまでなぞるように導くといった方法でプレイを行わせる. プレイ途中でカーソルが迷路の壁に触れてしまった場合は, 端末画面上にメッセージを表示するなどの方法で, ユーザに対してエラー通知が行われる. 図 10 に, クロスカーソルを用いたプレイ画面の拡大図を示す.

また, AR 技術による重畳表示をせず, 指で端末上の迷路を直接なぞるといった方法により, 撮影元の風景が目の前に無いような状況下でユーザの手元でプレイさせることも可能である. さらに, 生成した迷路は保存できるようになっており, 迷路画像を読み込むことで以前に生成した迷路を何度も繰り返しプレイすることが可能となっている. 図 11 に, スタートとゴールを設定した迷路の生成結果画像を示す.

4. 評価および考察

図 5 および図 9 の迷路生成時間・重畳画像の生成処理時間を比較すると, 迷路生成のもととなる素材画像の大きさが大きくなった場合, その迷路生成時間・重畳画像の生成処理時間も画像の大きさに応じて長くなっていることがわかる.

表 1 図 12 における迷路生成結果の比較.

	画像 (a)	画像 (b)
概形線抽出 (秒)	165	166
格子形成 (秒)	4	5
迷路生成 (秒)	21	36
セルの個数 (個)	1,491	2,246
辺の本数 (本)	3,375	5,103

図 12 および表 1 に, 素材画像の大きさが同じ (大きさ 800×600 ピクセル) であり, 画像の複雑さのみ異なる 2 種類の画像からそれぞれ迷路を生成した時の結果画像および各工程の実行時間, 生成されたセルの個数・辺の本数をそれぞれ比較した表を示す. 表 1 より, 画像の複雑さが変化しても概形線抽出・格子形成にかかる時間はほとんど変化しないが, 格子から迷路を生成する工程にかかる時間は素材画像が複雑なほど長くなるのがわかる. これは, 複雑な画像ほど形成される格子のセルの個数および辺の本数が増え, 格子構造と無向グラフの間の変換作業等にかかる時間が長くなるためである.

また, 最終的に生成されたアプリケーションを, 複数のユーザに実際にプレイしてもらうことで, アプリケーション自体の満足度やユーザインタフェースの操作性などの評価を行う予定である. また, Wong ら [2] が紹介している, 「迷路の難しさ」を解となる道の長さなどのパラメータから推定する方法を参考にし, 迷路を解くのにかかる時間と各パラメータの値の間の関係性を定式化することにより「難しさの調整機能」の実装などにもフィードバックを行っていく.



図 12 入力画像の大きさが同じで、画像の複雑さのみ異なる場合の迷路生成結果。図 (a) よりも図 (b) の方が撮影対象の個数が多く、複雑な画像となっている。

5. 結論と今後の課題

5.1 結論

携帯端末のカメラ機能を介して得られた入力画像と、生成した迷路を AR 技術を用いて重畳表示することにより、PC 上での実装を行っていた既存研究では実現しづらかった、現実世界が迷路になり、ユーザが手軽に没入感・臨場感を得られる「AR 迷路」という迷路の新しい楽しみ方をユーザに提供することが可能となった。

5.2 今後の課題

本研究で開発した迷路生成アプリケーションは、入力画像から特徴抽出を行うことで迷路生成および迷路の重畳表示を行うため、似たような色彩で全体的に特徴が乏しいような空間ではそれぞれの処理が行いづらい。したがって、このような特徴の乏しい空間においても各処理が行えるような汎用性のあるアプリケーションとするために、迷路生成の際、ユーザが手続的に迷路の構造を設定可能とするようなシステムの検討を行ったり、特徴の乏しい空間であると判断されたら迷路の重畳表示を現状のシステムであるビジョンベースの手法から GPS などの位置情報をもとにしたロケーションベースの手法を主体としたシステムに自動的に切り替えるような機能拡張を行うことで、より多くのユーザに広く楽しんでもらえるアプリケーションとしていくことが課題として挙げられる。

また、本アプリケーションでは、1 台の携帯端末上のみで全ての処理を行うものとなっている。現在の携帯端末にはネットワーク通信機能が搭載されており、近年、ネットワーク通信を利用した機能を搭載したアプリケーションも数多く開発されている。そこで、本アプリケーションにおいてもネットワーク通信機能を採用することで、ユーザが撮影した写真をネットワークを通して PC サーバ上に送り、迷路画像の生成処理など計算量の多い工程を処理能力の高い PC 端末で行い、その結果を再び端末で受け取るというクライアント・サーバ型システムにすることでさらなる処理能力の短縮を図ることが可能である。さらに、生

成した迷路をネットワーク上にアップロードする機能を搭載することにより複数のユーザで生成した迷路を共有するなどといった、ユーザ同士のコミュニケーションツールとしての役割を果たすことができるような、より魅力的なアプリケーションに拡張していくことが可能であると考えられる。

これらのような課題に対し、ユーザにとってより魅力的なアプリケーションとなるよう改良を加えていくことで、新たなスタイルのエンタテインメント創成につなげていくことが最終的な課題である。

参考文献

- [1] Shivers, O.: Maze generation (online), 入手先 (<http://www.ccs.neu.edu/home/shivers/mazes.html>) (2005).
- [2] Wong, F.J. and Takahashi, S.: “Flow-Based Automatic Generation of Hybrid Picture Mazes,” *Computer Graphics Forum*, Vol. 28, No. 7, pp. 1975-1984 (October 2009).
- [3] Kang, H., Lee, S., and Chui, C.K.: “Flow-Based Image Abstraction,” *IEEE Transactions on Visualization and Computer Graphics*, Vol. 15, No. 1, pp. 63-76 (February 2009).
- [4] Lowe, D.G.: “Distinctive Image Features from Scale-Invariant Keypoints,” *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91-110 (November 2004).
- [5] Bay, H., Tuytelaars, T., and Gool, L.V.: “SURF: Speeded up Robust Features,” in *Proceedings of the 9th European Conference on Computer Vision*, Vol. 1, pp. 404-417 (May 2006).
- [6] Rosten, E. and Drummond, T.: “Machine Learning for High-Speed Corner Detection,” in *Proceedings of the 9th European Conference on Computer Vision*, Vol. 1, pp. 430-443 (May 2006).
- [7] Calonder, M., Lepetit, V., Strecha, C., and Fua, P.: “BRIEF: Binary Robust Independent Elementary Features,” in *Proceedings of the 11th European Conference on Computer Vision*, Vol. 4, pp. 778-792 (September 2010).
- [8] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G.: “ORB: An Efficient Alternative to SIFT or SURF,” in *Proceedings of 2011 IEEE International Conference on Computer Vision*, pp. 2564-2571 (November 2011).
- [9] Kruskal, J.B.: “On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem,” in *Proceedings of the American Mathematical Society*, Vol. 7, No. 1, pp. 48-50 (1956).