

2020年度I類(情報系)コンピュータサイエンスプログラム卒業研究最終発表会

FPGAを用いたタイルドディスプレイシステムの ベゼルコレクション自動化

I類コンピュータサイエンスプログラム 学籍番号:1710203 成見研究室 木村智美

1 はじめに

1.1 背景

複数のディスプレイを用いて大画面を構成するタイルドディスプレイの普及が進んでいる。主に、駅構内などでのデジタルサインージとして用いられている。ただし、ディスプレイを縁取るベゼル部分にはコンテンツが出力されないために、画面数に合わせて単純な分割を行なっただけの画像や映像を表示するとずれているように見えるという問題点がある。この違和感を軽減するためにベゼルコレクションを行う。これは、使用するディスプレイが持つベゼルの幅を考慮して映像の一部を間引き、まるでコンテンツの上をベゼルが覆いかぶさっているかのように表示する手法である。

1.2 研究目的

本研究の目的は、FPGAを用いた4Kタイルドディスプレイシステムに対して、カメラを用いてベゼルコレクションの自動化を行うことである。ベゼルコレクションを行うためにはベゼル幅に応じた設定が必須である。従来のシステムでは、人の目で画面の表示状況を確認しながらコンテンツの拡大率や映像の位置を手動で調整する必要があった。この操作を自動化することによって、設置にかかる時間の短縮を実現する。

2 既存技術・既存研究

2.1 業務用マルチディスプレイコントローラーG413[1]

GeoBox G413は、タイルドディスプレイを構築するための業務用機器である。HDMI, DisplayPort, VGAのいずれかで入力した映像コンテンツを最大4つの表示機器に拡大分割表示することができる。25種類のプリセットパターンを内蔵しているほか、専用ツールを利用すれば任意のレイアウトで使用可能である。この機器でベゼルコレクションを行うには、レイアウト編集機能を用いてピクセル単位で水平・垂直方向に表示をずらす。調整は製品に付属しているリモコンを使用し、手動で行うため、1分程度かかる。

2.2 FPGAを用いた4Kタイルドディスプレイシステム [2]

白井は、FPGAを用いた4画面タイルドディスプレイシステムを開発した。低コストかつ軽量のFPGAとその拡張ボードのみで構成され、4枚のディスプレイを使用して

4Kの高解像度映像を表示することができる。それまでのFPGAによるシステムと違い、ベゼル幅の設定を変更するためにFPGA回路を再コンパイルする必要はなく、スイッチを用いて調整可能な動的ベゼルコレクション機能を備えている。

水平方向・垂直方向でそれぞれ拡大率の逆数を15bitの2進数で指定する。2つのスライドスイッチは下位12ビットのどの3ビットか指定し、4つのプッシュスイッチで拡大率を増減する。手動で設定するため、操作方法を理解した上で40~50秒程度かかる。

2.3 MovieTile[3]

太田と田中は、複数のモバイルデバイスを使用して自由な形状のマルチディスプレイ環境を構築するシステムを開発した。対応するアプリを起動したデバイスを並べ、隣り合うデバイス同士にピンチジェスチャーを行うと、それらの軌跡からデバイスのペアとその相対的な位置が検出される。これを、全てのデバイスが隣のデバイスと1つ以上の接続を持つように繰り返し行うことで、一つの大きなマルチディスプレイを構成することができる。直感的な操作で多様な画面配置を実現できる。ただし、画面間の同期に無線ネットワークを使う必要があり、ティアリングを防げない。

3 システム実装

3.1 システムの概要

図1に、本システムの概要図を示す。タイルドディスプレイを実現するためのFPGAボードは既存のシステムを用いた。これにカメラモジュールを備えたRaspberry Piを接続した。

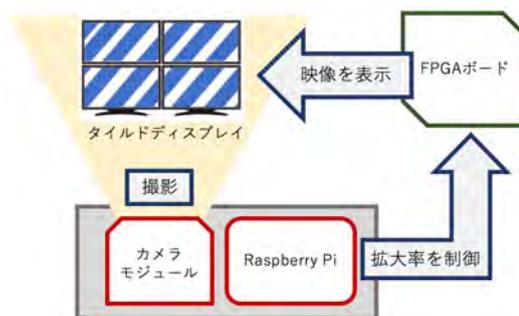


図1: システムの概要図

3.2 システムの使用手順

使用時にユーザーが行う手順を説明する。まず、FPGA ボードに入力機器および出力機器を接続し、タイルドディスプレイを構築する。本システムではベゼル幅および画面サイズが同一な4枚のディスプレイを格子状に並べることを前提とする。次に、タイルドディスプレイ全体にテスト用の画像を表示する。続いて、カメラを適切な位置に固定する。なお、あらかじめ Raspberry Pi では本システムのプログラムを実行しておき、フラットケーブルでFPGAと接続しておく。テスト画像が正しく認識できていれば、Raspberry Pi 本体に接続したLEDが点灯する。最後に、FPGA ボード上のスイッチを1にすると自動ベゼルコレクションが開始され、数秒で終了する。

3.3 システムの実装

ベゼルコレクションのプログラムは Raspberry Pi 上で Python と OpenCV を用いて実装した。プログラムの処理手順は以下である。

- (a) カメラから1フレーム画像を読み込む
- (b) 画像からオブジェクトを検出する
キャプチャした画像上の赤い部分に注目し、オブジェクトを検出する。オブジェクトとは、同じ色もしくは強度を持つ連続点を結ぶ曲線（輪郭）に囲まれた領域を指す。
- (c) 検出したオブジェクトをソートする
オブジェクトは、輪郭の座標として得られる。画面上に表示されている位置と対応するように座標の値に応じて並び替え、順にラベル番号0~9とする(図2左)。
- (d) オブジェクトから直線を検出する
ラベル番号0,2,3,4,5,6,7,9のオブジェクトの輪郭線に対してハフ変換を行なって直線を検出する。直線のパラメータ ρ と θ の値を確認し、必要とする直線(図2右)の式を決定する。
- (e) 対応する直線のズレに基づきFPGAへ信号を送る
対応する直線のパラメータを比較し、ずれている場合は拡大率を上げる。
- (f) 拡大率の変更対象のビットを下位に移動する
最も下位まで来たら終了する。

4 評価

開発したシステムによってベゼルコレクションのベゼル幅設定の操作が自動化できた。ベゼルコレクション設定後のタイルドディスプレイの画面は、設定前に比べて、ベゼル部分の表示ズレが修正され、画面を見たときの違和感は軽減している(図3)。



図2: オブジェクトのラベル番号のイメージ



図3: ベゼルコレクション設定前(左)と設定後(右)

表1: それぞれの処理1回あたりにかかる時間

	(a)	(b)	(c)	(d)	(e)+(f)
時間/msec	71.94	13.70	10.15	65.24	0.8378

ベゼルコレクションを行うのにかかる時間を10回ずつ計測した結果、明るい部屋では平均4.14秒、暗い部屋では平均5.79秒かかった。ただし、どちらの場合も10回のうち1回はベゼル幅の設定に失敗したため、原因を調査する必要がある。ディスプレイが発光していて周囲の明るさによる視認性の変化が小さいことから、部屋の明るさが本システムの性能に与える影響は小さいと考えられる。

また、プログラム内のどの部分にどれだけの時間がかかっているかを調べた。表1は5回ずつ計測した平均である。(a)~(f)は3.3節と対応している。この結果から、カメラからの画像取得と直線検出に特に多くの時間を費やしていることがわかった。これは、(a)ではカメラとの通信に時間がかかっていること、(d)では大量の計算を必要とするハフ変換をいくつものオブジェクトに対して行っていることが原因であると考えられる。

5 まとめ

本研究では、タイルドディスプレイに必要なベゼルコレクションの設定をRaspberry Piとカメラを用いて自動化した。ディスプレイ上の画像をプログラムで認識することで、人の目による設定作業がなくなった。本システムを使用することで3~10秒ほどでベゼルコレクションが行えるようになり、時間が短縮できる。これによって、ディスプレイの変更やタイルドディスプレイ全体の移動が今までより容易になったと言える。

今後の課題として、カメラの位置が正面でない場合やディスプレイが傾いている場合にも対応するため台形補正を実装すると、実用性が高まる。また、直線検出以外の方法を用いるなどして、さらなる高速化が実現できれば、移動する複数のディスプレイに対してベゼル補正をしながらコンテンツを表示することも可能になる。

参考文献

- [1] "GeoBox G413 多機能マルチディスプレイコントローラー", https://jimgs.jp/products/vns_g413.html, (最終アクセス日 2021年2月7日)
- [2] 白井暁, "FPGAを用いた4Kタイルドディスプレイシステムの開発", 電気通信大学 情報・ネットワーク工学専攻 修士論文, 2019
- [3] Takechi Ohta, Jun Tanaka. MovieTile: interactively adjustable free shape multi-display of mobile devices, SIGGRAPH Asia 2015 Mobile Graphics and Interactive Applications, Kobe, Article No. 18, 2015