

論文 / 著書情報
Article / Book Information

題目(和文)	Image SuperDistribution System: デジタル画像を媒体としたコンテンツ配信インフラ
Title(English)	Image SuperDistribution System: New content distribution infrastructure using digital images
著者(和文)	キムミンス
Author(English)	Minsu Kim
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第11292号, 授与年月日:2019年9月20日, 学位の種別:課程博士, 審査員:権藤 克彦,小池 英樹,渡部 卓雄,西崎 真也,齋藤 豪,比嘉 邦彦
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第11292号, Conferred date:2019/9/20, Degree Type:Course doctor, Examiner:,,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

東京工業大学大学院

博士論文

Image SuperDistribution System:
New content distribution infrastructure
using digital images

(Image SuperDistribution System:
デジタル画像を媒体としたコンテンツ
配信インフラ)

指導教員 権藤 克彦 教授

平成31年9月

提出者

専攻 情報理工学院
情報工学系
情報工学コース

氏名 Minsu Kim

Image SuperDistribution System: New content distribution infrastructure using digital images

指導教員 権藤 克彦 教授

情報理工学院 情報工学系
情報工学コース
Minsu Kim

Content Providers (e.g., Disney, Sony), who make content, distribute their own content through Content Platformers (e.g., Netflix, Amazon) on the mobile devices for the security and platform dependency reasons. One of the reasons why Content Providers use Content Platformers is that it is required to not only build and deploy a native application of iOS and Android to play and save content, but also make users download it, to distribute digital content on the mobile. However, it is not easy for all the Content Providers to develop a native application, so that they still have been using Content Platformers. As a result, users have to use Content Platformers' applications, and Content Platformers take a revenue share, which adversely affects the profit of Content Providers. To distribute content without Content Platformers and Content Providers' platforms, the most important thing is that a platform-independent material is required to connect a user to content. Basically, the distributed content cannot be saved without a native application on the mobile, because the content file itself cannot be saved without an encryption logic on the user's device for a security reason. To break the rule, the new platform-independent material needs to be developed to save the content without an application. It should be intrigued to a user, easy to be integrated, have no dependency on Content Platformers and cheap. In this paper, a new content distribution model is proposed using digital images, which is Image SuperDistribution System (ISDS). The model

uses digital images as content distribution media with a new color Codemark, Robust Index Code (RIC), for this use-case, which is also proposed on this paper. RIC is proposed as a brand new robust Codemark targeting digital images so that it makes possible to link content with a digital image. Until now, the main purpose of Codemark is to distribute digital content using offline media. Due to the main purpose of Codemark, it cannot be used on digital images. It has a high robustness on only printed images. Hence, RIC is designed with high robustness on JPEG Compression and Resize targeting digital images compared to existing researches, such as QR Code. In addition, RIC embeds a remote database index to digital images so that users can reach to any digital content from images. Experimental results, using RIC encoder and decoder, have shown high robustness on JPEG Comp. and Resize of the RIC. The embedded database indexes can be extracted 100% on 960 pixels and PSNR 16 above. RIC images can be used as a platform-independent material and new content distribution model ISDS can be realized. In ISDS, Content Providers upload their content to our implemented content pool console of an open content server or a Content Provider's server. Subsequently, a RIC image and a URL, linked with the uploaded content, will be generated. Users can easily access the content clicking the URL or decoding the RIC image with the RIC SDK. Not only a URL and an image are platform-independent materials but also SuperDistribution infrastructure is open dissemination of content so that this system does not depend on specific applications or Content Platformers on the distribution process. Of course, only a URL may be used for the content distribution, but a URL is easily mistaken and it is hard to imagine to manage all the content as URLs which he or she bought. Hence, saving content as a digital image into the device gallery makes a user manage all the content and open them easily. Finally, the new system achieves increasing user's convenience and independence from Content Platformers.

目 次

第1章	序論	1
1.1	研究背景	1
1.2	従来研究	5
1.3	研究目的	6
1.4	論文の構成	12
第2章	背景と目的	13
2.1	問題提起	13
2.2	提案手法と目的	25
第3章	関連研究	28
3.1	画像にデータを埋め込む技術	28
3.2	コンテンツ配信	38
第4章	Robust Index Code	40
4.1	Robust Index Codeの必要性	41
4.2	Robust Index Codeの実装	43
4.3	Robust Index Codeのデザイン改善	56
第5章	Image SuperDistribution System	66
5.1	ビジネスモデルとは?	67
5.2	ビジネスモデルの例	70

5.3	Image SuperDistribution System の概要	73
5.4	デジタル画像の欠点	76
5.5	SDK の配布	77
5.6	Image SuperDistribution System の構造	79
5.7	アプリケーション構築	84
5.8	コンテンツ配信プロセス	88
第 6 章	予備実験	92
6.1	Robust Index Code の予備実験	93
第 7 章	結論	102
7.1	まとめ	102
7.2	研究の貢献	104
7.3	今後の課題	105
	謝辞	107
	参考文献	107
付録 A	ユーザーの動画再生フロー	122
付録 B	電子透かし実装コード	129
付録 C	ISDS コンソール利用シーン	133

第1章

序論

1.1 研究背景

近年デジタルコンテンツの消費は激しく増加し、その市場の大きさは2017年で2兆9,633億円を占める [1]。しかし、そのコンテンツの配信インフラには嬉しくない問題がある。Warner Bros.、Paramount、Disney、Sonyのようにコンテンツを制作及び配信を行うコンテンツプロバイダー（本論文ではプロバイダーと呼ぶ）は、Netflix、Hulu、Amazonのようなコンテンツプラットフォーム（本論文ではプラットフォームと呼ぶ）を通じて、コンテンツを配信している。また、モバイル環境で配信するコンテンツは、iOSやAndroidのモバイルOSを通して配信する。しかし、モバイルOSとプラットフォームを通すコンテンツ配信モデルは、プロバイダーにとって嬉しくない。なぜなら、Figure 1.1で示す通り、コンテンツの収益が多く分配されるためである。

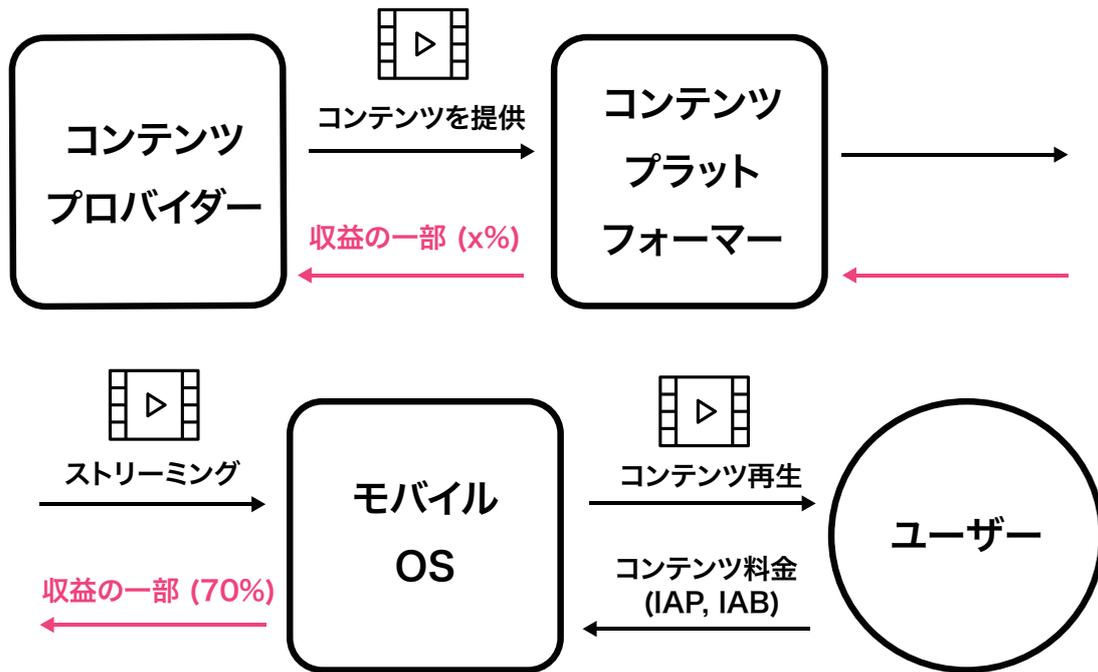


Figure 1.1: As-is: コンテンツ配信フロー [2]

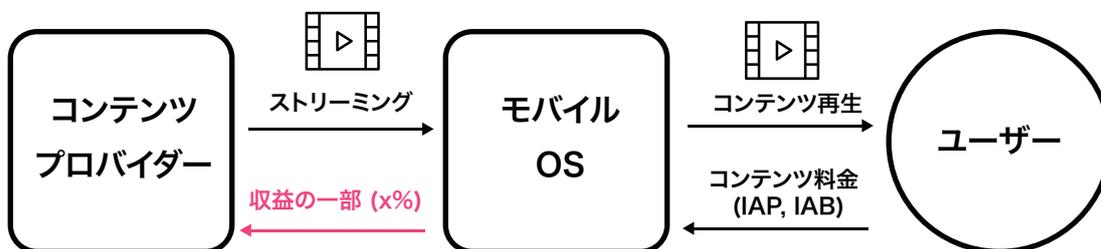


Figure 1.2: To-be: 本研究のコンテンツ配信フロー [2]

既存コンテンツ配信モデルには大きく二つの問題が存在する。一つは、コンテンツがモバイル環境で消費されればされるほど、AppleのiOSやGoogleのAndroidのモバイルOS側に力が集約することである。コンテンツがモバイルOSで消費されるため、モバイ

ル OS を作った企業は収益分配率を自分たちで決められる。例えば、Apple はアップルストアとデジタルコンテンツのトランザクションから 30% の手数料を取り、一年に 100 億ドル以上の売上を上げている [3]。そのため、プロバイダーは多くの収益を得られていない。

二つ目は、プラットフォーマーの存在である。モバイルデバイス上では、有料コンテンツを消費するためには、セキュリティ問題によりネイティブアプリケーションが必要になる。モバイルウェブブラウザ上でもコンテンツの消費自体は可能ではあるが、DRM 付きのコンテンツを再生可能にする Web API である Encrypted Media Extensions (EME) API がモバイルブラウザではサポートされていない場合がある。特にブラウザアプリよりもアプリ内ウェブビューでサポートされていない。従って、有料コンテンツを配信するためにネイティブアプリケーションの開発が必要であるが、全てのプロバイダーがアプリケーションの開発、デプロイ、プロモーションすることは困難である。そのため、プロバイダーはプラットフォーマーにコンテンツを提供し、プラットフォーマーが全てのコンテンツを自分のアプリケーションで配信する。プラットフォーマーはいわゆるユーザーとプロバイダーの間に存在する中間レイヤーと言える上、プロバイダーの収益は

もっと減っていく。例として、あるプラットフォームは45%を手数料として取っている [4]。その結果、プロバイダーはプラットフォームへの依存性を恐れ、人気作や最新作はプラットフォームに提供していなく、全てのコンテンツ提供をやめるプロバイダーも存在する [5]。

上記で述べた問題を解決するためには、Figure 1.2で示す通り、プラットフォームがないオープンなコンテンツ配信インフラが必要である。具体的には二つの解決策がありうる。一つはプロバイダーが自らプラットフォームを作り上げ、直接コンテンツを配信することである。実際はDisneyはNetflixへの配信をやめ、自分たちのプラットフォームを準備している [5]。しかし、開発及び運用には専門の技術者が必要であり、膨大なコストがかかってしまう。二つ目はプラットフォームに依存しない媒体を利用したコンテンツの配信である。プロバイダーが媒体とコンテンツを紐付け、直接コンテンツを配信できるモデルである。超流通 [6]は理想的なコンテンツ配信モデルであり、この解決策の実装モデルとして目的がマッチする。しかし、調べた限り超流通を実現できたシステムを未だ存在しない。そこで本研究はデジタル画像を媒体とした超流通システムを提案し、コンテンツ配信モデルの改善を試みる。

1.2 従来研究

コンテンツ配信は主にビジネスのサービスとして展開している。そのため、コンテンツの配信自体の研究は少なく、コンテンツの著作権保護に関する研究が多い。

コンテンツの著作権保護にはほとんど Digital Rights Management (DRM) 技術を使用する [7, 8]。コンテンツを暗号化し、暗号キーは検証されたユーザーのみに期限付きで配布することで、コンテンツの著作権を保護している。暗号化したコンテンツを他人に共有しても、暗号キーがないと再生できないため、有料コンテンツの課金モデルを守ることができる。Netflix、Hulu、Amazon のような大体のプラットフォームは DRM 技術を用いて、コンテンツ配信を行っている。

コンテンツ配信モデルの研究はバーコード [9], 電子透かし [10] などを用いたデバイス間で直接コンテンツを共有する研究が存在する。例えば、バーコードをデバイス間の通信設立に用い、デバイス間で直接コンテンツを共有する形の研究が存在する [9]。また、[11] では Microsoft Foundation Classes (MFC) 上の API Hook を用いた DRM システムを提案している。コンテンツをセキュアに配信することが可能になるが、コンテンツを Eメールや P2P コネクションなどによるコンテンツ転送を基盤としているため、イン

ターネット上でコンテンツ配信が容易でない。その上、MFCはモバイル環境では使用することができない。本研究は1対1ではなく、プロバイダーが不特定多数のモバイルユーザーにコンテンツ配信できる環境構築を目標とする。

1.3 研究目的

1.3.1 コンテンツ配信モデル

本研究は下記に述べる特徴を持つ新しいコンテンツ配信モデルを提案する。

1. プロバイダーが直接コンテンツプロモーションできる。
2. プラットフォーマーへの依存性がない配信モデルである。
3. コンテンツをどのプラットフォームにも依存しない独立した空間に保存する。
4. ユーザーはいつでもコンテンツに再アクセスでき、コンテンツを管理しやすい。

本研究は超流通システムにデジタル画像を媒体として取り入れた、Image SuperDistribution System (ISDS) を提案し、ISDSのモデルの確立及び実用化を目指したシステムの実装を目的とする。

ISDSのデザインキーポイントは、コンテンツの保存場所にある。既存では、ユーザーが購入したコンテンツをプラットフォームのアプリケーションに保存し、そのアプリ上で再生できる仕組みになっている。しかし、どれかのアプリケーション上でコンテンツを保存するということは、そのアプリケーションにとってユーザーのアクセスが増え、利益になってしまう上、アプリへの依存性も高まる。従って、本研究はコンテンツを、デバイスギャラリーのように、どのプラットフォームにも依存しない独立した場所に保存する必要がある。その結果、プラットフォームに依存しない場所に、あらゆるコンテンツの保存を可能にする媒体の選別が重要になる。既存でプラットフォームに依存しないメディアはURLと画像の二つだけ存在する。それらは特別なアプリケーションを必要とせず、あらゆるプラットフォームで消費できる。既にURLはQRコードに埋め込まれ、コンテンツ配信手法として用いられている。しかし、URLは一文字間違えるだけでコンテンツへアクセスができなくなるほど、間違いやすいメディアである。そのため、本研究はURLとデジタル画像両方をコンテンツ配信媒体として用いる。URLはユーザーがSNSやE-mailなどのインターネットサービスからコンテンツへアクセスしやすくするため、ユーザーは最初URLを通じてコンテンツへアクセスし、コンテンツの保存

には画像を用いる。画像はデバイスのギャラリーに保存され、いつでも再度アクセスできる。画像はURLとは異なり、ユーザーが間違えることがなく、画像のみから紐づくコンテンツを想像できるなど、コンテンツ配信に最も適した媒体だと言える。

そこで、デジタル画像からコンテンツへのアクセスする方法として、デジタル画像を対象とする新たなコードマーク、Robust Index Code（以下、RIC）を提案する [12, 13, 14]。RICは、インターネット上での画像拡散において、ロバスト性が非常に高いコードマークであり、Facebook、Instagram、LINEなどに画像をアップロードしても埋め込んだデータが破壊されない。そして、コンテンツ実体はインターネットのデータベースに保存し、そのインデックスを画像に埋め込むことにより、様々なデータと画像を紐づけることが出来る。従って、画像さえあればコンテンツへアクセスできる環境を構築できる。本研究で実装したRIC SDKを個々のアプリやデバイス自体の出荷段階で埋め込むことにより、全てのユーザーが新たなアプリケーションのインストールなしで、URLやRIC画像からコンテンツを再生できる。

本研究はISDSを実現するため、コンテンツプールコンソールを実装し、RIC SDKを埋め込むため、通信キャリアやスマートフォン制作会社にコンタクトした。プロバイダーはコンテンツをコン

ソールにアップロードし、値段や再生可能デバイス台数などのコンテンツを利用規約を設定すると、コンテンツと1対1で紐づくURLとRIC画像を得られる。ユーザーはプロバイダーが共有したURLや画像からコンテンツへアクセスできる。

収益に関しては、モバイルOS上のデジタルコンテンツの購買は彼らのアプリケーションレビューガイドラインにより、必ずiOSのIn-App Purchase (IAP)、AndroidのIn-App Billing (IAB) を利用しなければいけない [15, 16]。ユーザーは既に持っているはずのAppleやGoogleのアカウントで簡単にコンテンツを購買でき、ISDSはトランザクションの情報とデバイスの固有IDを保存することで、コンテンツの購買情報をユーザーの会員登録やログインなしで管理できる。また、プロバイダーは30%のIAPとIABの手数料を除く残り全ての収益を得られる。

1.3.2 Robust Index Code

1.3.1節で述べた通り、本研究はデジタル画像とコンテンツを紐づけるために新たなコードマークであるRobust Index Code (RIC) を提案する。RICは唯一デジタル画像を対象としたコードマークとなり、ISDSの実現に必要不可欠と言える。

画像を埋め込む技術を提案する関連研究には二種類存在する。

一つは画像に色を用い、生データを埋め込むコードマーク研究、もう一つは画像の見えない部分にデータを埋め込み、著作権判別用途で主に使われている電子透かし (Digital Watermark) 研究である [17]。

コードマーク研究には、DENSO社が提案した白と黒を用いて生デジタルデータを画像に直接埋め込むQR Code [18]がある。QR Codeはコードの向きと位置を認識するための三つの四角、データの情報を埋め込んでいる四角、誤りを訂正するために四角の三つのパーツで構成している。印刷画像に対し、誤りが少なく、URLなどのテキスト情報を容易に埋め込むことにより、広く使われている。NurwonoらのCQR Code [19]は白黒だけで表現するQR Codeを、色のレイヤーを置き、埋め込めるデータ量を3倍に増加した。また、Microsoft社は4つの色の三角形にデータを埋め込むHCCB [20]を提案した。これ以外にも、20種類以上のコードマークが提案されている。しかし、従来のコードマークはすべて印刷画像へ描画し、カメラで読み取ることを目標としているため、インターネット上の画像への描画に対し、ロバスト性が弱い。そのため、本研究では既存研究とは異なり、新たにインターネット上の画像に対応しているコードマークを提案する。

電子透かし研究には、Jianshengが提案したDCTとDWTを合わ

せた Watermarking 方式 [21] や、Barni らが提案した DCT を用いた Watermarking 方式 [22] など、様々な方式で研究が進んでいる。電子透かしはデジタルデータの著作権を守ることが目的のため、人が認識しにくい画像部分にデータを埋め込む。そして、埋め込んだデータが元のデータとどれほど一致しているかにより、著作権が付いたデータであるか判断する。そのため、電子透かしで埋め込まれたデータは必ずしも 100% で元のデータを抽出する必要がなく、実際にも抽出されない。すなわち、電子透かしでデータを埋め込むと、コードマークと同様に、埋め込んだ生のデータをそのまま誤りなく抽出することができなく、本研究が目標としている、100% 抽出可能な画像にデータを埋め込む技術として応用できない。本研究はデータの Robust 性を目標としている部分が既存の電子透かし研究と大きく異なる。

RIC は、リサイズと JPEG 圧縮に対し、非常に高いロバスト性が保てるように設計した。JPEG 圧縮によるピクセルの変化をなるべく防ぐことを第一として捉え、OpenCV を用いて、埋め込む、読み取ることができるライブラリを実装し、有用性の実験を行った。

1.4 論文の構成

本稿は2章で研究の背景及び目的を述べる。3章では関連研究として、コードマーク・電子透かし研究、コンテンツ配信研究を述べる。4章では本研究で提案する Robust Index Code について説明し、5章では Image SuperDistribution System について述べる。6章では有用性を示すための実験について述べる。7章では本研究のまとめと貢献、今後の課題を述べる。

第2章

背景と目的

コンテンツプロバイダーがコンテンツプラットフォームを通過せずコンテンツを配信できる共通のインフラシステムを実現することは難しい。全てのプロバイダーがコンテンツに対するセキュリティ担保や自分たちの配信環境を整えることが困難なため、プラットフォームを通過しないでコンテンツを配信することができない。その結果、プラットフォームへの依存性、収益の分配など、様々な問題が発生している。そこで本研究では、理想的な配信インフラモデルである超流通モデル [6] を採用した、デジタル画像を用いたコンテンツ配信モデルを提案する。

2.1 問題提起

本研究には二つの大きな問題の解決に取り組んだ。一つはプロバイダーが手軽に自分のコンテンツを配信できる共通インフラの

不在である。もう一つは、画像劣化に耐えられるデジタル画像を対象としたコードマークの不在である。この章では不在の原因を述べる。

2.1.1 超流通

超流通システム [6] は1983年に筑波大学の森亮一によって提案された、デジタルコンテンツ流通の理想の概念である。超流通は次の三つの原理で定義される [23]。

- システムによるデジタルコンテンツの保安ラッパーはコンテンツが複製される時も残る。
- コンテンツ提供者が指定する利用条件のもとでコンテンツ利用者はいつでも利用できる。
- コンテンツ提供者が指定するコンテンツ利用料をコンテンツ利用者から支払えるシステムが必要である。

超流通はプロバイダーとユーザーに対し、一番理想的なコンテンツ配信モデルである。超流通による配信の元では、コンテンツの利用条件はプロバイダーが決められる。コンテンツ自体は高いセキュリティのレイヤーで守られ、違法にコンテンツを抜き取ったり、コピーすることが容易でない。ユーザーは追加のハード

ウェア・ソフトウェアなしでコンテンツを再生できる。

理想である超流通システムは、本研究で調べた結果、未だに完全なるシステムとして実現できていない。ただ、大きな企業がプロバイダーとユーザーから収益を得ることを目的とした自分たちの配信環境を構築している。例として、Appleのアプリストアは全てのトランザクションから30%の手数料を取ることで、1年で100億ドルの大きな収益を得ている [3]。

2.1.2 コンテンツ配信インフラの不在

超流通モデルは特にモバイル環境上（主にAppleのiOS、GoogleのAndroid）のデジタルコンテンツ配信において、実現がもっと難しくなる。基本的にモバイル環境において暗号化したコンテンツの再生・保存及びセキュアな決済を行うには、ネイティブアプリケーションが必要となる。なぜなら、ブラウザやウェブビューのWeb APIは暗号化したコンテンツの再生や保存をサポートしていない。最近Encrypted Media Extensions (EME) APIが開発され、PCのブラウザではDRM付きのコンテンツが再生できるが、モバイルブラウザやアプリのウェブビューでは未だにサポートされていない。また、会員登録やログインなしでウェブ上では同一ユーザーであることの認識を保証できない。CookieやFingerprint

[24] 方式を用いてユーザーのトラッキングがある程度はできるが、100%の保証ができない。その結果、プロバイダーが自分のプラットフォームを開発しなければいけないが、容易ではないため、プラットフォームマーがプロバイダーからコンテンツを集め、自分のアプリケーションを通じて配信を行う。

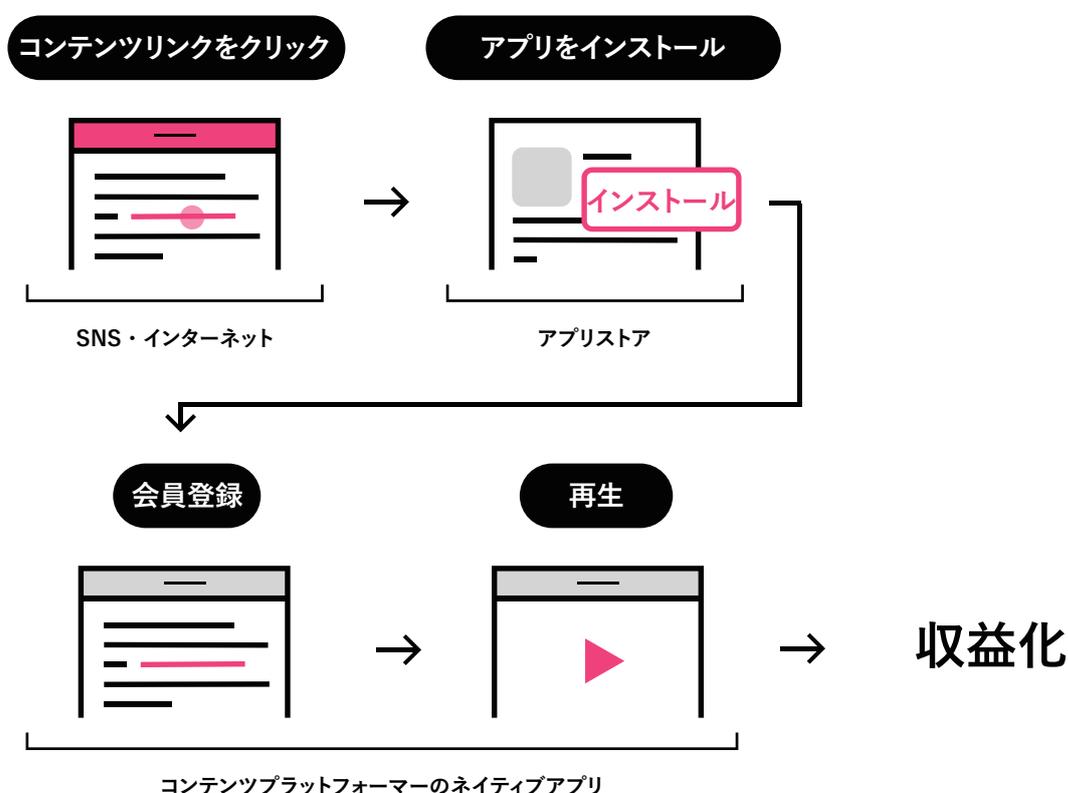


Figure 2.1: As-is: コンテンツ広告から再生までのユーザーフロー

Figure 2.1 はインターネット上のコンテンツへ繋げる広告リンクを押したところから、実際のコンテンツの再生までのユーザー

フローを表す。いかにプロバイダーが自分のコンテンツの広告をインターネットに掲載しても、ユーザーは最初必ずアプリをインストールしなければならない。そのユーザーフローの例を付録Aに示す。従って、プロバイダーはプラットフォーマーへの依存性が高くなり、パワーバランスが崩れ、プラットフォーマーは自分たちでコンテンツの利用条件を決めたり、高い収益配分率を要求することが可能になる。例えば、AmazonはKindleの収益配分を著者側に35%と70%で提供するが、70%はe-Bookのファイル転送料を払わせるなどの条件が付く[25]。Youtubeは広告収益を作成者側に45%を収益として還元する[4]。このように、技術的限界と新規ユーザー獲得の難しさにより、プロバイダーがプラットフォーマーへ依存してしまう問題（例えば、配信料や配信戦略の決定に裁量権が少ないこと）を本研究では依存性問題又は従属性問題と呼ぶ。その例として、アメリカではTV利用者の77.8%がSVOD（月額制：Subscription Video On Demand）に加入、そのうち37%がNetflixに加入しているため[26]、既にプロバイダーがプラットフォーマーに大きく依存していると推測できる。

上記で述べた理由により、プロバイダーは最新又は人気のコンテンツ（以下、トップティアコンテンツ）をプラットフォーマーに配信したがる。その上、戦略的に自らコンテンツ配信を行い

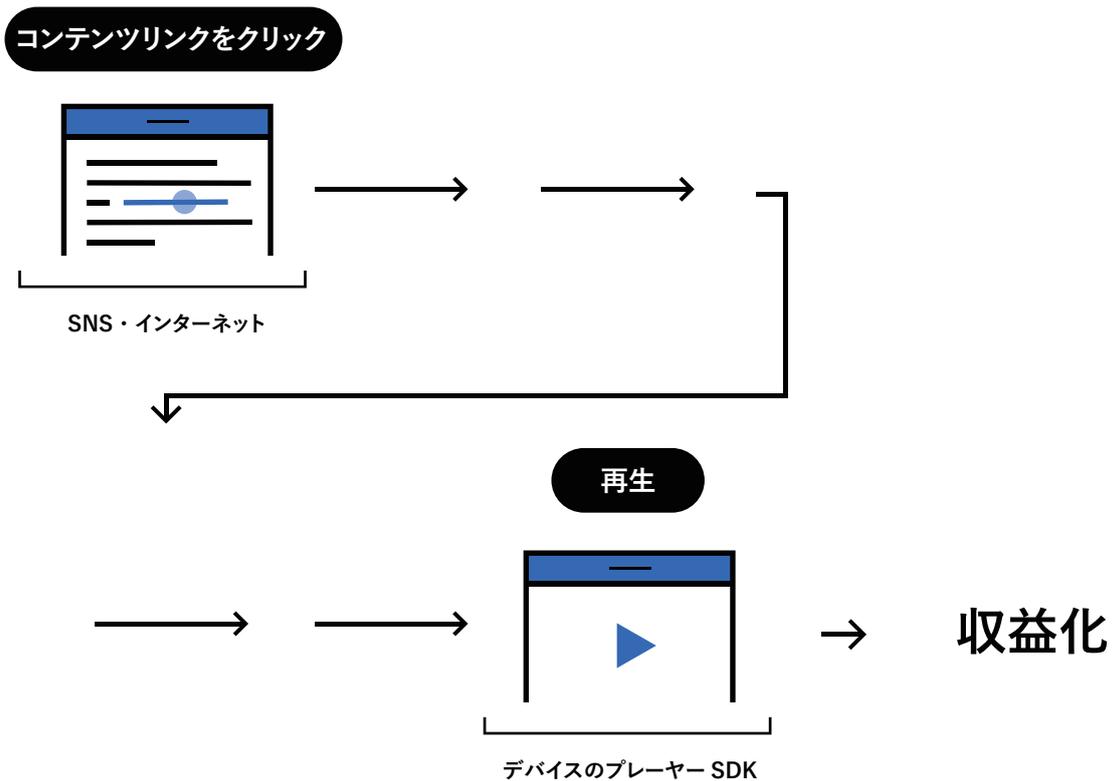


Figure 2.2: To-be: コンテンツ広告から再生までのユーザーフロー

たく、自分たちでプラットフォームを作ろうとしている。例えば、Walt Disney はアメリカやカナダでオンラインストリーミングに対しトッププラットフォーマーにも関わらず [27]、これ以上 Netflix にコンテンツを配信しないと決定し、Disney+を準備している [5]。また、WarnerMedia も同じく Netflix への配信をやめ、自社プラットフォームに移行している [28]。Walt Disney や WarnerMedia は世界トップの大きな会社であるため、プラットフォームを作る能力を持っているが、中小のプロバイダーはそうでない。従って、本

研究ではFigure 2.2で表す新しいコンテンツ配信モデルの提案を目的とする。

2.1.3 動画配信サービス調査

本研究で提案する新しいコンテンツ配信モデルが、既存動画配信インフラにおいてどれほど影響を与えられるかを確認するため、既存動画配信サービスを調査した。現在、動画配信ビジネスモデルのトレンドは月額制である。Netflix、Hulu、Amazon Prime Video、U-NEXTなどでは、ユーザーが1ヶ月月額を購入すると、プラットフォームのすべてのコンテンツを再生できる。従って、コンテンツの単価が低くなり、薄利多売が続いている。

プロバイダーは上記の問題を解決するため、自らプラットフォームの展開を試みている。TBS、テレビ東京、WOWOWのParavi [29]、フジテレビのFOD [30]、東映アニメーションの東映アニメオンデマンド [31]が例となる。しかし、モバイル環境では新たな数多くのアプリケーションをインストールしなければいけなく、新規ユーザーからのコンテンツ再生まで道のりが容易でない。その結果、数十個存在する月額制動画配信サービスのどれも、2018年動画配信市場の推計によると、15%以上の利用率を達成できていない [32]。

従って、新たなプラットフォームによるコンテンツ配信サービスの成功は容易ではなく、一つのプラットフォームという技術的限界が存在する。そこで、新たなアプリケーションのインストールを必要としない、プラットフォーム独立型コンテンツ配信システムを用いると、現状打破できると予測される。本研究はその部分に着目し、超流通を導入して問題の解決を試みる。

2.1.4 コンテンツ配信インフラの課題

本研究ではFigure 2.2のように、プロバイダーが直接ユーザーにコンテンツを届けられる環境構築を目標とする。プロバイダーが目指す理想の環境は、既存サービスの調査により、以下を満たすものであると推測される。

1. コンテンツをユーザーに直接販売したい。
2. プロバイダーがコンテンツの広告を掲載すると、ユーザーがその場ですぐ再生できてほしい。
3. プラットフォーマーへの依存性をなくしたい。
4. コンテンツ配信におき、他のサービスが配信以外の目的で利益になることは望ましくない。
5. ユーザーはいつでもコンテンツに再アクセスできてほしい。

しかし、上記の項目を満たす環境の構築には、社会的な課題と技術的な課題が存在する。まず、社会的な課題は以下の通りである。

1. 新たなプラットフォームに見えるサービスには依存性問題がある。
2. コンテンツをアプリケーション内に保存することも依存性問題がある。
3. 著作権問題により、サービスにはプロバイダーのものであることを掲載する必要がある。

上記の社会的な課題により、独自のモバイルアプリケーションを用いたインフラを提案することはできない。従って、下記の通りの技術的な課題が発生する。

1. アプリケーションなしで暗号化したコンテンツをどう再生するか？
2. 暗号化したコンテンツはどこに保存するか？
3. 保存したコンテンツをどう再閲覧させるか？
4. アプリケーションなしで決済はどう行うか？

本研究は上記の問題を解決するためには、新たな媒体が必要だと認識し、デジタル画像を媒体とすることを提案する。即ち、コ

コンテンツをデジタル画像に紐付け、デジタル画像だけあればいつでもどこでもコンテンツにアクセスできるインフラを意味する。

2.1.5 コードマークの問題

2.1.4節の最後に述べたとおり、デジタル画像をコンテンツと紐づけることが本研究の課題である。デジタル画像はデジタルカメラやスマートフォンの開発により、使用頻度が増加している。インターネットサービスである Facebook には毎日 3.5 億枚のデジタル画像がアップロードされる [33]。しかし、新聞や雑誌などで使用する印刷画像も健在しているため、画像にデータを埋め込む技術は主に印刷画像がメインターゲットとなっている。

印刷画像をデジタル世界と繋ぐため、QR Code、CQR Code、HCCB など、画像にデータを埋め込むコードマーク技術が存在する。既存のコードマークは 2 次元バーコードであり、QR Code は白黒、HCCB は 4 色の色を使い、画像上に描画する。画像には生データを暗号化し埋め込み、ユーザーはコードマークの写真を撮り、機械で読み取ることで埋め込んだデータを見ることができる。コードマークを用いると、デジタルプロダクトを印刷画像に埋め込めるため、O2O 戦略 (Online to Offline) として注目を浴びている。

しかし、コードマークをデジタル画像にロバスト性を保って埋め込む研究は行われていない。インターネット上で画像とテキストをアップロードすると、その紐付けは他のプラットフォームに移る瞬間、切れてしまう。そのため、画像自体にテキストなどのデータを埋め込むことにニーズが存在する。それにもかかわらず、コードマークがデジタル画像へ導入されないことには、コードマークが画像の劣化に弱く、特にJPEGで影響を受けやすい問題が存在するためである。JPEGは画像形式の一つであり、類似する部分を圧縮する。その特徴により、画像のファイルサイズを他の形式より小さくできるため、インターネットサイトの72.5%が使用するほど最も使われている [34]。しかし、画像圧縮の影響により、JPEGにコードマークを埋め込むと、埋め込んだデータが破壊されやすい。具体的には以下で述べる問題点が存在する。

- Facebookなどのインターネット上のサービスに画像をアップロードすると、ファイルサイズを低くするため、JPEG圧縮やリサイズが発生する。例えば、1280pxの幅で232KBのファイルサイズを持つ画像をFacebookに投稿すると、35KBのファイルサイズに自動で圧縮される [35]。コードマークは色でデータを埋め込むため、JPEG圧縮などの画像劣化により、色が変わり、データが変形される。すなわち、ロバスト性が低くな

り、埋め込んだデータを100%読みとることができない。しかし、印刷画像の場合、色が変わるなどの劣化現象が起きにくいため、この問題を解決しようとする既存研究はない。

- 既存コードマークは画像に生のデータを直接埋め込むため、コードマークを読み取るプログラムがあれば、誰でも埋め込んだデータを閲覧できる。そのため、閲覧させたいユーザーを限定できない。QR Codeは広告などに主に使われているが、ユーザーを限定できると、プライベート用途で使うことも可能になる。
- 既存コードマークは画像に埋め込めるデータ量に限界が存在する。データの量は、QR Codeは最大約3KB [18]、CQR Codeは最大約9KBである [19]。そのため、主にテキスト、URL、位置情報が埋め込まれ、画像や音楽などのデータを埋め込むことはできない。コードマークのサイズを大きくすると、埋め込めるデータ量も増えるが、画像を妨げることになりうる。コードマークのサイズを小さくしつつ、データ量を増やすことが難しい。

2.2 提案手法と目的

本研究は、2.1節で述べた、既存で存在するコンテンツ配信環境の改善を試みる。プラットフォームの依存性をなくし、プロバイダーが直販できる共通インフラを構築するには、2.1.4節で述べたプロバイダーの要求項目を満たし、技術的及び社会的課題を解決できる新たな手法が必要となる。

本研究は2.1.1節で取り上げた超流通を、デジタル画像を媒体としたコンテンツ直販インフラモデルとしての提案を目的とする。従来、超流通を既存プラットフォーム上で動作するように実装できた事例は、調べた限り存在しない。本研究ではその点に着目し、既存プラットフォーム上で、プラットフォームに依存しないメディアを活用することにした。そのメディアはデジタル画像である。デジタル画像に直接データを埋め込み、読み取ることができれば、プラットフォームに依存しないコンテンツの媒体として利用できる。

しかし、デジタル画像を対象としたコードマークを提案するには、2.1.5節で述べた問題を解決する必要がある。従来、画像にデータを埋め込む研究には印刷画像を対象としたコードマーク、著作権を守ることが目的である電子透かしが存在する。しかし、本研究が解決しようとしている問題は、二つの分類に属さない。

そのため、新たなコードマークデザインが必要となる。そこで本研究では、以下の性質を持つ、デジタル画像を対象としたコードマーク、Robust Index Code (RIC) [12, 13, 14] を提案する。

- 1) 画像形式：インターネット上で最も使われている JPEG 形式を対象とする。
- 2) ロバストなコードマーク：インターネット上のあらゆるサービスで行われる画像のリサイズや JPEG 圧縮に耐えられ、ユーザーが 100% データを読み取れる。
- 3) データリンク：テキスト、リンクにとどまらず、画像や音楽などの様々なタイプのデータをリモートデータベースに保存し、インデックスを画像に埋め込むことで、あらゆるコンテンツと画像を紐づけられる。
- 4) 閲覧制限：データを埋め込む原作者は閲覧対象にパスワードなどの制限をかけることができる。

最後に、デジタル画像にデータを埋め込み、読み取れる技術が完成しても、実際コンテンツをどう配信するかのアーキテクチャーが必要である。2.1.4 節で述べた問題を解決しつつ、プロバイダーがコンテンツをアップロードし、ユーザーが再生・課金するまでの全体のフローの構成も本研究の目的とする。従って、本研究は

大きく二つのモデル（ロバストなコードマークとコンテンツ直販モデル）を提案し、有用性を示すことを研究目的とする。

第3章

関連研究

この章では本研究に関連する既存研究について述べる。本研究では大きく二つの研究目的を持つため、関連研究も二つの部類に分けて述べる。3.1節では画像にデータを埋め込む既存研究の紹介、3.2節では超流通の関連研究を紹介する。

3.1 画像にデータを埋め込む技術

画像にデータを埋め込む技術は大きく3つの方式に分けられる。本章ではそれぞれのカテゴリーの関連研究について述べる。

1. 主に印刷画像を対象にし、画像の中でユーザーが認識する部分にデータを埋め込むコードマーク研究
2. 主にデジタル画像を対象とし、画像の中でユーザーが認識しにくい部分にデータを埋め込む電子透かし研究

3. 主にデジタル画像を対象とし、画像に何らかのデータを埋め込み、元の画像を復元するリバーシブルデータ隠し研究 (Reversible Data Hiding, RDH)

3.1.1 コードマーク研究

画像にコードマークを貼り付ける、もしくはコードマークのみを印刷し、ユーザーをオフラインからオンラインに移す目的としてコードマーク研究が行われている。コードマークは大きく三つに分類される。

(1) 1次元バーコード:

横方向にだけ情報を埋め込むコードマークであり、黒の四角や線に情報を埋め込む。主にキャッシュレジスター（通称レジ）などの機械から読み取る [36]。

(2) 2次元モノクロバーコード:

縦横方向に情報を埋め込むコードマークであり、白と黒の四角で情報を埋め込む。主に携帯電話やカメラで読み取る [37]。

(3) カラーバーコード:

縦横方向に情報を埋め込むコードマークであり、最大8色

を使い、四角や三角で情報を埋め込む [38]。

(1) には30種類以上のバーコードが存在する [36]。そのうち、Pitney Bowes 社が開発し、主に図書館や血液銀行で使用した Codabar [39] (Figure 3.1)、IBM社が開発し、スーパーマーケットで使用した Universal Product Code [40] (Figure 3.2) が代表的な例である。Figure 3.1 や Figure 3.2 の例の通り、1次元バーコードは主に線にデータを埋め込む。Codabar は最大15桁、Universal Product Code は12桁の数字を埋め込むことができるが、データ量を増やすには、横向きに全体のコードマークサイズを増やす必要があり、データのロバスト性を増やすには、縦向きに全体のコードマークサイズを増やす必要がある。そのため、コンパクトなサイズのコードマークには向いていない。



Figure 3.1: Codabar の例 [36]

(2) には20種類以上のバーコードが存在する [36]。Figure 3.3 に



Figure 3.2: Universal Product Code の例 [36]



Figure 3.3: QR Code の例 [36]

示す、Denso社が開発したQuick Response Code (QR Code) [18]は、位置を認識するための三つの大きな四角とデータを埋め込む白黒の小さい四角で構成される。2Dの縦横方向でデータを埋め込むが、位置や方向を認識するための大きな四角を置くことにより、速い読み取り速度を可能にしている。また、小さい四角は生データを埋め込む四角と、誤り訂正用の四角に分け、カメラによる画像のぼやけの対策をしている。QR Codeは印刷画像を対象にしているため、デジタル画像上でのみ発生するリサイズやJPEG圧縮などの劣化の対策はされていない。そのため、位置認識用の四角が劣化されやすく、読み取れなくなる。RICはデジタル画像を対象にし、リサイズやJPEG圧縮に対しロバスト性が高い事を保証する。

(3)の代表的な例はMicrosoft社が開発したHigh Capacity Color Barcode (HCCB) [20]である。Figure 3.4に示した通り、4色または8色の色を用い、三角形にデータを埋め込むカラーコードである。QR Codeの最大データ量が入るサイズの場合、HCCBは最大データ量約7KBとなる。Figure 3.5に示す、NurwonoらのColor Quick Response Code (CQR Code) [19]はQR Codeをカラーコードに拡張したものである。QR CodeをR、G、Bの各チャンネルに埋め込む形となり、縦横に加え、深さを入れた3次元バーコードと

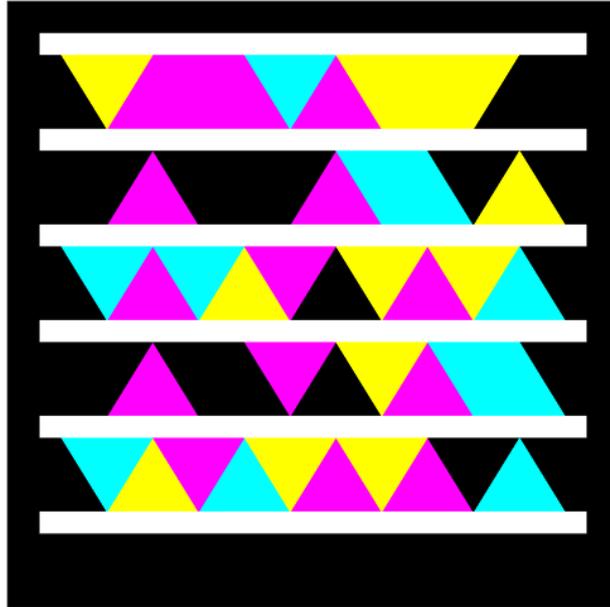


Figure 3.4: HCCBの例 [36]

なる。3つのチャンネルにQR Codeを描画するため、最大データ量は約9KBとなる。一定の画像サイズではCQR Codeが最大データ量が大きい。しかし、カラーになった分、画像劣化の影響を受けやすくなり、デジタル画像の劣化とはさらに相性が悪い。既存研究はQR Codeのデータ量の改善を目標としていることに対し、RICはデジタル画像を対象にし、画像劣化にロバスト性を高めることを目標にしているため、目標が大きく異なる。そのため、本研究のコンテンツ配信モデルはモバイルというデジタル上での配信となるため、デジタル画像の劣化に耐えられるRICのみが選択



Figure 3.5: CQR Code の例 [19]

できる唯一のコードマークとなる。

3.1.2 電子透かし研究

電子透かしはコードマークとは目的が異なる画像にデータを埋め込む技術である。インターネット使用率の増加にあたり、画像自体の著作権を保護を目的として提案された研究である。即ち、デジタル画像を対象としているため、各電子透かし研究は、デジタル画像の劣化に耐えられることを実験として行う上、電子透かし自体がJPEG圧縮、ガウス雑音 (Gaussian Noise)、切り取

り、リサイズなどに高いロバスト性を持つことが証明されている [41, 42, 43]。

電子透かしは、画像を周波数空間に変換し、人の目が認識しにくい中・低周波数部分にデータを埋め込む手法を取っている。JPEG 圧縮が同じく、画像の周波数空間の中・低周波数のデータを削るが [44]、電子透かしはその削除に耐えられるよう工夫し、強いロバスト性を持つように構成されている。最終的に、画像から埋め込んだデータを抽出し、本来のデータとの一致率を計算することで、画像の著作権を認める仕組みとなる。



Figure 3.6: DCT 電子透かしの例 [45]

電子透かしにはいくつかの手法が存在するが、人が認識しにくい部分の抽出やデータの暗号化が電子透かし研究キーポイントとなっている。

(1) 一つの変換を用いる研究

離散コサイン変換 (DCT) や離散ウェーブレット変換

(DWT) などの一つの変換を用いて目が認識しにくい部分を抽出する。

(2) 複数の変換を用いる研究

(1)の変換方法と主に特異値分解 (SVD) を合わせ、目が認識しにくい部分を抽出する。主に、電子透かしが埋め込まれた画像が原本画像のように綺麗に見えることが目標とされている。

各方式共に電子透かしの挿入による画像の損傷をなるべく防ぐ、埋め込んだデータ（主に画像）がなるべく劣化されないように工夫することを目的としている。しかし、電子透かしはその目的から、埋め込んだデータを100%そのまま抽出する必要性はなく、著作権を認められるデータの一致率以上のようになるよう、埋め込んだデータを抽出できれば良い。即ち、既存の電子透かし研究はデータの劣化しすぎを防いではいるが、100%そのままの抽出は目標としていない。[46]でもJPEG圧縮を受けた電子透かしは91%の一致率を実験結果として提示している。本研究のコンテンツ配信モデルは、デジタル画像をコンテンツと紐づけることが必要なため、画像とコンテンツが必ず1対1の関係を保証しなければならない。従って、画像から抽出したデータは本来のデータと完全一致する必要がある。[47]の研究が電子透かしを元にし、インデッ

クスを画像に埋め込み、抽出することを提案しているが、JPEG圧縮によりデータの損傷が確認されている。その結果、本研究ではJPEG圧縮に耐えられる新たなコードマークのRICを提案する。

3.1.3 リバーシブルデータ隠し研究

リバーシブルデータ隠し研究 (Reversible Data Hiding, RDH) はデジタル画像を対象とし、画像にデータを埋め込み、元どおりに画像を抽出することを目的とする。様々な手法のRDHの研究が存在するが、デジタル画像の劣化に実用的に耐えられる手法は未だ実装できていない。[48] ではDC係数を用いたヒストグラム修正によるRDH手法を提案している。データの埋め込みによる画像の損傷をさせず、ファイルサイズの増加も低くすることに焦点を当てられているため、強いJPEG圧縮の実験が行われていない。[49, 50, 51] は画像に暗号キーを埋め込む画像暗号化手法をRDH原理で提案しているが、画像劣化を検討していない。[52] ではヒストグラム、[53] ではModification of Prediction Errors (MPE)を元にしたRDHを提案しているが、どれも画像劣化のロバスト性に関して実験されていない。従って、画像劣化に対し高いロバスト性を持つRICが本研究の目指すコンテンツの媒体として相応しい技術である。

3.2 コンテンツ配信

近年コンテンツ配信は、バーコード [9], 電子透かし [10], DRM [7, 8]などの手法を用い、急速な成長を見せている [7]。しかし、コンテンツ配信において理想な形である「超流通システム」と呼ばれる、プラットフォームに依存しないコンテンツ配信手法は未だに実現できていない。

[9] はバーコードを用いたデバイス間の通信によるコンテンツ転送を提案しているため、インターネット上のコンテンツ配信の手法としては適切でない。[10] は電子透かしを用いたRDHによる画像著作権を提案しているが、画像以外のコンテンツの保護はできない。

その結果、DRMはコンテンツ配信において主に使用される技術である [7, 8]。オリジナルコンテンツを暗号化し、検証されたユーザーのみに暗号キーを与え、モバイルアプリケーションなどでコンテンツの復号化を許す手法である。暗号化されたコンテンツはURLを用いるか直接転送が必要であり、モバイルアプリケーションなしでインターネット上でコンテンツを配信することが困難である [54]。[11] では Microsoft Foundation Classes (MFC) 上の API Hookを用いたDRMシステムを提案している。コンテンツをセキュアに配信することが可能になるが、コンテンツをEメール

やP2Pコネクションなどによるコンテンツ転送を基盤としているため、インターネット上でコンテンツ配信が簡単でない。その上、MFCはモバイル環境では使用することができない。本研究で提案するコンテンツ配信モデルは、上記の問題を解決し、インターネット上で簡単にコンテンツを配信できるモデルである。

我々が調べた限りでは、デジタル画像を用いたコンテンツ配信の既存研究、サービスは存在しない。コンテンツ配信に関する特許 [55, 56, 57] が存在するが、上記の特許はDRM環境の中、コンテンツを購入した一人のユーザーが、購入したコンテンツを他のユーザーへ共有する方法に集中している。即ち、ファイルをモバイル環境で直接共有は困難であるため、モバイル環境でのコンテンツ配信の研究は極めて少ないと言える。そこで、本研究はコンテンツの共有をデジタル画像という媒体を用いることで、モバイル上での新たなコンテンツ配信システムを提案する。

第4章

Robust Index Code

本研究は新たなコンテンツ配信モデルにおいて、配信の媒体としてデジタル画像を用いる。プラットフォームに依存しないメディア二つのうち一つであるデジタル画像を活用することで、コンテンツを簡単に配信・共有できる。デジタル画像をコンテンツと紐付ける方法が問題となるが、デジタル画像を対象とした中のデータを100%抽出できる既存研究は存在せず、本研究では新たなコードマークを提案し、Robust Index Code (RIC) と命名する [12, 13, 14]。RICは、インターネット上で自動で発生するリサイズやJPEG圧縮に対し、非常に高いロバスト性を持つコードマークとなる。本章では、RICがコンテンツ配信モデルにおいての必要性、RICの提案、RICのデザイン改善に関して述べる。

本研究ではRICライブラリをOpenCV [58] を使い、iOSやAndroidにデプロイ可能なC++で実装し、Objective-C++とAndroid NDK

[59]を用いてモバイル環境へのポーティングも実装した。4.1節ではRICの必要性、4.2ではRIC v0.1の提案及び実装、4.3節ではRICのデザイン改善について述べる。RICの開発環境をTABLE 4.1に示す。

TABLE 4.1: 開発環境

モデル	MacBook Pro (13-inch, 2017)
CPU	3.5 GHz Intel Core i7 Dual Core
メモリ	16 GB 2133 MHz LPDDR3
ストレージ	APPLE SSD AP1024J 1TB
OS	macOS 10.14 Mojave
Xcode	Xcode 10.1 (10B61)
OpenCV	OpenCV C++ Library 2.4.13

4.1 Robust Index Codeの必要性

2.1節で述べた通り、既存コンテンツ配信はプラットフォームに依存する形である。特にモバイル環境上では、暗号化したコンテンツを復号化するにはネイティブアプリケーションが必要になり、アプリケーションの開発・運用・広告などはプロバイダーに負担がかかる。そのため、プロバイダーは最新作や人気作をプラットフォームにおいての配信をしたがらなく、我々の調べた限り、

プラットフォームのサービスでは最新作のコンテンツを見る
ことができない。そこで、ユーザーとプロバイダーを直接繋げられ
る新たなインフラが必要となる。

2.1.4節で述べたプロバイダーが目指す理想の環境を構築するた
めには、コンテンツ配信においてネイティブアプリケーションの
必要性を断ち切ることが最も重要である。アプリケーションが必
要なくなると、プラットフォームへの依存性、プロバイダーの
アプリケーション開発必要性が自然となくなる。即ち、アプリな
しでコンテンツへアクセスできる配信の媒体が必要になる。現在
のモバイルOSでプラットフォームに依存しないメディアはURL
と画像のみが存在する。URLでもインフラの構築は可能だが、
URLは一文字間違えるだけでコンテンツへのアクセスができなく
なる上、数が多くなると整理が困難になる。また、URLのみでは
どのコンテンツへアクセスできるか想像しにくい。そこで、本研
究は画像をコンテンツへアクセスできる媒体として用いることに
した。画像はインターネット上からユーザーが簡単に保存でき、
保存場所も独立された空間であるスマホのアルバムに保存され、
いつでもコンテンツへ再アクセスが可能になる。また、画像には
訴求力があり、画像を見るだけで中のコンテンツが想像できる。
それでは画像からコンテンツへアクセスするにはどうすれば良い

だろうか。

本研究はインターネット上のあらゆる環境でコンテンツが配信できるよう、画像自体にデータを埋め込むコードマーク方式を選択した。メタデータや電子透かしなど、画像にデータを埋め込む方法はいくつか存在するが、3.1.1節で述べてように、本研究のニーズと合致しない。また、既存のコードマークはデジタル画像を対象にしていなかったため、既存技術でインターネット上の配信は不可能である。そのため、本研究はデジタル画像を対象とする新たなコードマークを4.2節で提案する。

4.2 Robust Index Codeの実装

本研究は極めて高いRobust性を持つ新しいカラーコード、Robust Index Code (RIC) を提案する。RICのアプローチは、インターネット上で起きる画像のリサイズやJPEG圧縮に耐えられる新しいコードマークデザインを提案する。100%のRobust性を満たすことにより、生のデータとコードマークの間にインターネット上のデータベースを置くことが可能になる。すなわち、ユーザーが埋め込みたいデータはデータベースに保存し、そのインデックスを画像に埋め込むことができる。従って、リンクさせるデータ量には制限がなく、様々なデータをリンクできる。

4.2.1 Robust Index Code デザイン

本研究の目標はインターネット上で最も行われる画像劣化、画像のリサイズ、JPEG圧縮により [41]、埋め込んだデータが破壊されないコードマークの提案である。QR Codeなどの既存研究が上記の画像劣化により、データが破壊される理由は、データを埋め込んだピクセル（データピクセル）が他のピクセルにより変化が生じるためである。こうした問題はJPEG圧縮の量子化ステップが原因である [60]。JPEG圧縮は画像8x8ピクセル単位でDCT変換を行い、量子化ステップで類似する部分を削除する。従って、データを埋め込んだピクセルが周りのピクセルとともに一元化されてしまう。その問題を解決するためには、データピクセルを十分大きく取り、データピクセル同士の距離も十分離れる必要がある。そこで、本研究はFigure 4.1に示す通り、ビットデータを埋め込むデータピクセルを半径 r の円とする。また、二つの円の距離は d ピクセル離す。データを埋め込む場所はCQR Codeと同様である [19]。一つの円には3ビットデータを埋め込む。Rチャンネル、Gチャンネル、Bチャンネルにそれぞれ1bitを、0なら D_0 色、1なら D_1 色を描画することでデータを埋め込む。Figure 4.1は D_0 が80、 D_1 が255の例である。

もう一つ考えるべきポイントは画像からの侵食である。既存

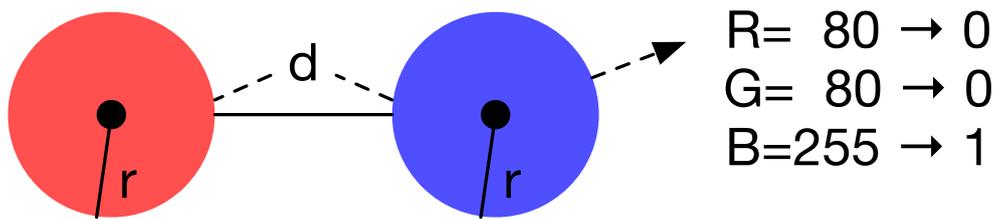


Figure 4.1: RIC データピクセル [12]

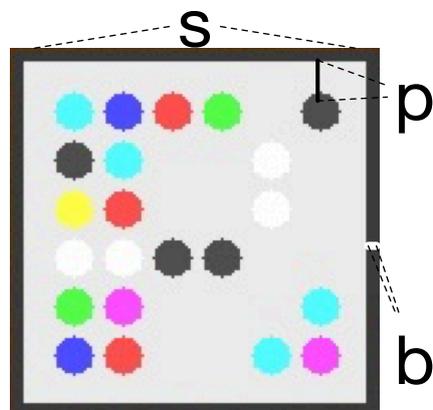


Figure 4.2: RIC デザイン v0.1 のコードマーク部 [12]

コードマークとは違い、RICは既に存在する画像に埋め込むことを想定している。そのため、画像からの侵食を無視できない。そこで補正する部分が、背景とボーダーである。データを埋め込む円をマトリックス構造で並べ、そのマトリックスの背景色をグレーにする。また、マトリックスに黒いボーダーを付けることにより、画像からの侵食を防ぐ。ボーダーは色よりも太さが重要である。ボーダーが1ピクセルの太さであると、画像ピクセルより



Figure 4.3: RIC デザイン v0.1

劣化される可能性があるが、一定以上の太さを持つと、画像からの影響は考えにくくなる。 r を4、 d を2、 D_0 を80、 D_1 を255に仮定したRICデザインの例をv0.1としてFigure 4.2、Figure 4.3*に示す。

RIC v0.1において、埋め込めるデータキャパシティは、ボーダーを外した大きい四角の一辺の長さを s 、画像からコードマークへの

*画像を引用：<https://unsplash.com/photos/fr6em8x4f2I>

パディングを p 、円の数を N_c とした時、次の通りとなる。

$$capacity(\text{bit}) = 3N_c \quad s.t. \quad 0 < N_c \leq \left(\frac{s+d-2p}{2r+d}\right)^2$$

Figure 4.2 では 22 個の円が存在する。従って、66 ビットのデータを埋め込むことができる。Facebook に毎日約 3.5 億枚の画像がアップロードされるため [33]、サービス化には 50 ビット以上あれば妥当だと考えられ、66 ビットのキャパシティは十分と考えた。

RIC v0.1 の実装には OpenCV C++ Library 2.4.9 [58] を導入した。OpenCV はインテルが開発・公開した画像処理・画像解析および機械学習等の機能を持つライブラリであり、様々なプラットフォームにサポートしている。RIC の描画は OpenCV のメソッド (例: `cv::rectangle`, `cv::circle`) を用いると簡単に実装できる。本研究では綺麗なコードマークを描画するため、入力画像の小さい辺を 1280px にリサイズしてから RIC を描画する。

4.2.2 Robust Index Code 読み取り

この節では、RIC の読み取る部分について述べる。RIC の読み取りは QR Code と同じく、三つのパートで構成されている。RIC の発見、データの抽出、抽出したデータの検証である。

コードマークの発見

RICは画像からボーダー付きの四角を発見することにより、コードマークを見つける。しかし、四角を見つけることが容易ではない。そこで、ハイパスフィルタとロウパスフィルタを適用すると、画像から面積のある領域を見つけることができる。その領域は、本来4.2.1節で描画した通り四角であるが、画像のリサイズやJPEG圧縮などの画像劣化により、四角の四つの辺がFigure 4.4のように凸凹になってしまい、OpenCVのエッジ検出メソッドで簡単に四角だと判断することが難しいからである。また、四角の中には複数の反対色の円が存在するため、提供されているAPIを利用して判断することは難しい。そのため、本研究では四角発見アルゴリズムを提案・実装した。直線をつなぎ合わせると四角になることに着目し、凸凹になる部分や途中円が入る部分を許しながら、同じ色の直線をつなぎ合わせ、四角を発見する方式である。画像に元々RICと似たような四角がある場合、RICだと判断してしまうリスクが生じるが、RICではない四角の区別をデータ検証段階で行い、間違った四角を発見する事を防ぐ。

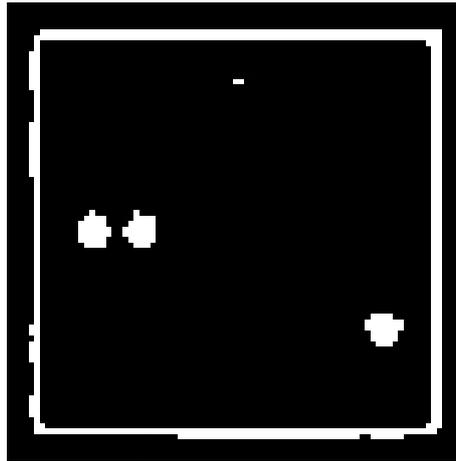


Figure 4.4: ロウパスフィルタを適用した画像の一部 [14]

データの抽出

RICからデータを抽出するには、まずデータが埋め込まれている円の位置を抽出する必要がある。画像からコードマークを発見すると、その四角の長さの比率を計算する事により、RICで描画したすべての円の中心を計算する事が出来る。しかし、その中心は理想の中心であり、実際の中心とはズレが発生する。RICを埋め込んだ画像はインターネット上でリサイズが起きると、円の中心はリサイズの比率分移動するが、その際小数点ができるので、描画される際、切り上げまたは切り下げが発生し、元々の円とズレが生じる。そのため、本研究ではHough Circle Transformを用い、円の理想の中心を実際の中心へ移し、ズレを補正する。

実際の中心を見つけると、そのデータ円から埋め込んだ3ビット

トのデータを抽出する。円のうち、JPEG圧縮により一番劣化されない場所は、周りから一番離れている中心となる。そのため、中心の色が D_0 に近いと0、 D_1 に近いと1に変換する。しかし、色が曖昧な時は0か1か区別することができない。そこで本研究はデータ候補リストの概念を導入し、色の判断が難しい場合は0と1両方を候補リストに追加し、本来のデータが必ず候補リストの中に入るようにする。さらに、候補リストのサイズには N_d の限界値を置き、候補リストが膨大になる場合は、劣化しすぎた画像と認識し、データ抽出をやめることとする。データの抽出が完了すると、検証段階によって、候補リストを絞る作業を行う。

4.2.3 Robust Index Codeの検証

コードマークから抽出したデータ候補リストは、画像劣化の際変化が生じる色から抽出したものであるため、間違っただータが抽出されている可能性が存在する。そのため、抽出したデータが正しい埋め込んだデータであるか確認するステップが必要となる。最終的には候補リストから正しい一つのデータのみを残すことを目的とし、複数のデータが得られた場合はユーザーに選択させる。本研究ではチェックサムとヒストグラムアルゴリズムを用いてデータ検証を行う。

チェックサム

チェックサムは誤り検出符号の一つであり、信頼性は低いですが、簡単なアルゴリズムにもかかわらず、一般的に99.5%以上の検出率がある [61]。検出方法は、元々のデータから算出したチェックサムと、ダメージを受けたデータから算出したチェックサムの一致を確認することで誤りを検出する。チェックサムの算出方式は割り算のあまり、ワード列の個々のワードの総計など様々であるが、どの方式も非常に簡単である。

本研究ではインデックスをある定数で割ったあまりをチェックサムとして利用している。例えば、あるインデックスが $281474976710653_{(10)}$ とする。割る定数を $60000_{(10)}$ だとすると、チェックサムは次のようになる。

$$\begin{aligned}Checksum &= 281474976710653_{(10)} \% 60000_{(10)} \\ &= 50653_{(10)} \\ &= 1100010111011101_{(2)}\end{aligned}$$

RIC v0.1 は Figure 4.2 で示した通り、66ビットを埋め込める。そのうち、50ビットをデータ、残り16ビットにデータのチェックサムを埋め込む。上記の数式で計算したチェックサムを16ビットに変換して埋め込む形となる。しかし、チェックサムもRICの円に

埋め込むため、画像劣化に対し、データだけでなく、チェックサム自体も変わってしまう問題が生じる。そのため、画像ヒストグラムを用い、RICデータ検証の信頼性を向上する。

画像ヒストグラム

画像ヒストグラムは画像の特徴を表すヒストグラムの一種であり、画像の色合いの数を度数分布グラフ（以下、グラフ）として表現したものである。本研究ではヒストグラムの例として Figure 4.5の画像を使用する。この画像のBチャンネルから抽出したヒストグラムを Figure 4.6の青いグラフに示す。ヒストグラムの x 軸は 0~255の色の値であり、 y 軸はピクセル数となる。Figure 4.5の画像は赤色が多く、青色は黒色に使用されているため、そのBチャンネルヒストグラムは x 値が低いところに集約されるグラフとなる。

ヒストグラムは、本来の画像のヒストグラムと、劣化された画像のヒストグラムは幾何学的に似ている特徴を持つ [62]。そのため、ヒストグラムを画像の識別子として利用する研究は既に行われている。Xiangらのヒストグラムを利用した画像のハッシュ値 [62] はヒストグラムグラフを 0、1のバイナリー列として変換し、ハッシュ値として利用する。この手法は再現率が高いが、適合率が低く、ヒストグラム全体のデータを保存するため、保存すべきデー



Figure 4.5: ヒストグラムの元画像

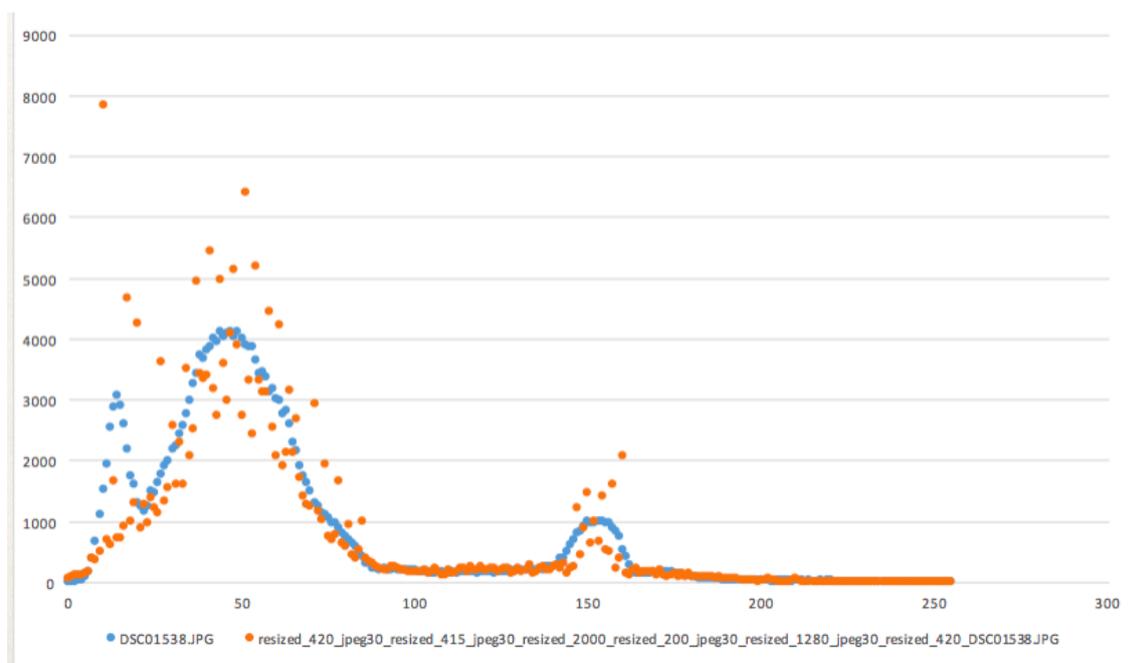


Figure 4.6: Figure 4.5のBヒストグラム [14]

夕量が多くなる。

本節では画像を対象にし、再現率と適合率が共に高い、ヒストグラムを利用した検証アルゴリズムを実装する。この検証により、画像の激しい劣化による間違っただデータの抽出を防ぐことが可能になる。さらに、悪意のあるユーザーがRICを偽造した際、そのアクセスを防ぐことも可能になる。RICは画像のリサイズとJPEG圧縮の劣化を対象にしている。そのため、ヒストグラムも二つの劣化を対象とする。Figure 4.5にリサイズとJPEG圧縮を複合した10個の画像劣化を与え、ヒストグラムを比較するグラフをFigure 4.6のオレンジ色のグラフに示す。グラフに示されている通り、画像劣化後のヒストグラムは図形は崩れるが、頂点は変わらないことが発見できる。そのため、本研究ではヒストグラムの頂点の情報を識別子として用いる。

本研究で提案するヒストグラム識別子算出アルゴリズムを次に述べる。

- 1) 画像を基準サイズにリサイズし、 m 分割する。ヒストグラムは色の位置を考慮せず、色の数だけ計算するため、画像そのままからヒストグラムを算出すると、違う画像も似た画像として判断する可能性が高くなり、適合率が低くなる。そのため、画像を分割し、区域ごとにヒストグラム比較を行う。
- 2) 分割したそれぞれの区域をブロックとする。ブロックをRGB

チャンネル分けを行う。ヒストグラムはグレースケールのみ描画できるため、チャンネル分けが必要となる。

- 3) ブロッグの一つのチャンネルに Histogram Equalization を適用する。このメソッドは画像のコントラストを調整するものであり、画像の特徴を強調する。
- 4) ヒストグラムを算出し、 n 個の頂点を y 軸の値を大きい順で抽出する。

次に、比較手法について述べる。RIC はデータとコードマークの間に、インターネット上のデータベースを置く。そのため、検証に必要なデータを画像に直接ではなく、データベースに保存することも可能になる。従って、上記で算出した頂点情報をデータを挿入するデータベースに付随して追加し、そのインデックスを RIC で埋め込む。その結果、画像からインデックスを取り出す際、データベースへアクセスし、そこから本来の画像から算出したヒストグラム情報を元に検証を行うことが可能になる。そのフローは次の通りである。

- 1) 劣化画像からインデックスを読み取り、データベースから本来の画像のヒストグラムデータをダウンロードする。
- 2) 劣化画像からヒストグラム情報を算出する。

- 3) 二つのヒストグラムデータから、頂点の x 軸値が同じである頂点の数を数え、区域ごとの一致率を計算する。
- 4) 総 m 個のブロックの一致率のうち、一番低い一致率を比較の一致率と選択し、60%以上であれば同じ画像と判断する。

画像10,000枚に6つの画像劣化、リサイズ700px、リサイズ700px × JPEG 50% 圧縮、リサイズ700px × JPEG 30% 圧縮、リサイズ400px、リサイズ400px × JPEG 50% 圧縮、リサイズ400px × JPEG 30% 圧縮を適用し、総60,000枚の画像で検証の再現率と適合率を求めた。実験の結果、適合率は94.3%、再現率は99.9%、F値は0.970となり、有効性のある検証方法だと言える。

4.3 Robust Index Codeのデザイン改善

4.2節で提案されたRIC v0.1はデジタル画像を対象にし、JPEG圧縮やリサイズの劣化に高いロバスト性を持つことにより、本研究で求める画像技術のニーズに適合する。コンテンツはインターネット上のデータベースへ保存し、そのインデックスをRICとして画像に埋め込む。閲覧の際は、インデックスを読み取り、紐づいてるもとのコンテンツへアクセスができる。従って、RICのみが独立したコンテンツ配信の構築を可能にするとと言える。しかし、RIC v0.1は下記の問題を持つ。

1. コードマークが画像の上に埋め込むため、画像の著作権を侵害する。
2. デザイン質が高いポスター画像を対象にすると、コードマーク自体のデザインと相性が良くない。
3. 画像が不正修正された場合の対策が出来ていない。

上記の問題を解決するため、RICをコンテンツ配信に適したデザインに改善が必要である。理想的には、Figure 4.7で拡大している、コードマークを囲むグレーのボックスを取り払いたいですが、技術的に困難な要素が存在し、その理由を下記に示す。

1. 画像にRICが存在するか判断できない。
2. 画像の中で、RICの位置を把握できない。
3. データ円をJPEG圧縮から守ることができない。

本研究では、上記の困難さを、電子透かしを導入し、下記に示す通り解決を試みた。

画像にRICが存在するか判断できない

RICを埋め込む画像に電子透かしを埋め込むことで、RICの存在を把握する。電子透かしを用いてデータを正確に抽出することは出来ないが、本来の電子透かしの目的の通り、画像

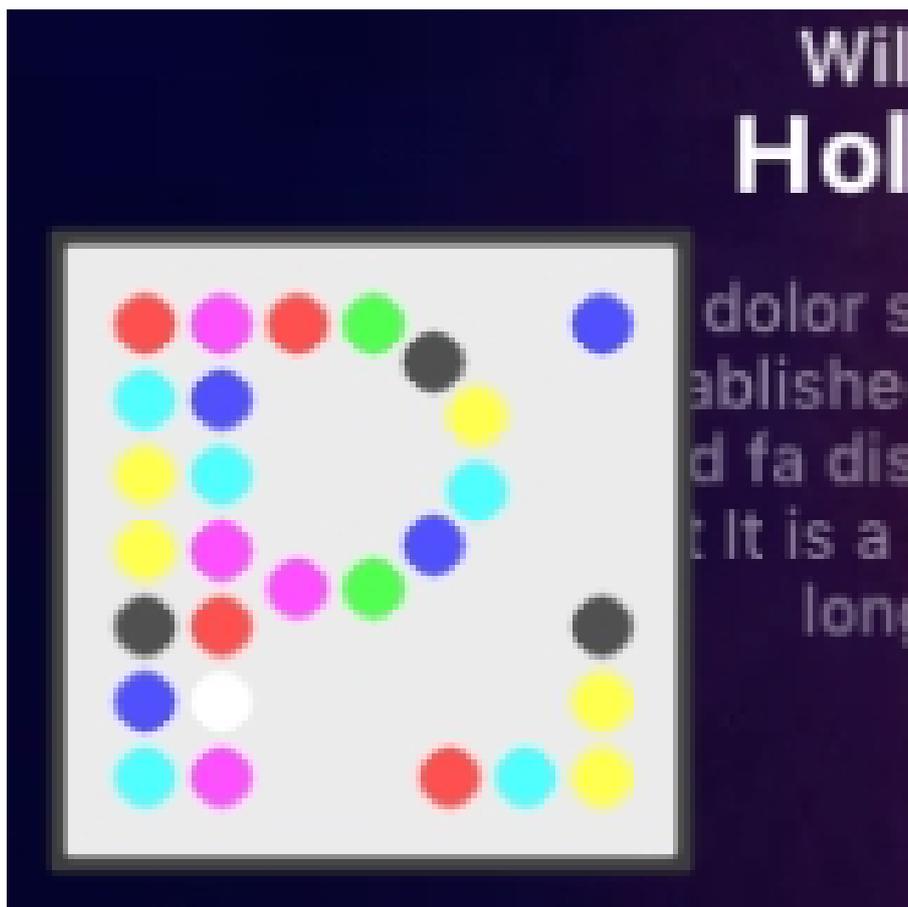


Figure 4.7: Figure 4.3でコードマーク部分を拡大した画像

にデータの埋め込みの痕跡を探すことには高いロバスト性を持つ。そのため、RICの埋め込み時に電子透かしも埋め込むことで、RIC存在の判断が可能になる。

画像の中で、RICの位置を把握できない

RIC v0.1デザインにおいて、画像の中からRIC位置の抽出を実装した経緯は、画像の上にコードマークを埋め込むため、画像がクロップされたり、変形されても抽出可能にするため

の配慮であった。しかし、本研究のコンテンツ配信において画像著作権問題により、画像が修正された場合はコンテンツへのアクセスを防がなくてはならなくなった。そのため、RICの位置判断が不可能になったこと及びRICの位置修正を不可能にするため、RICの位置を固定にした。RICの位置が不変になると、電子透かしの読み取りが成功すると、すぐにコードマークのデータ抽出作業に入るため、時間短縮にも繋がる。

データ円をJPEG圧縮から守られない

グレーボックスをなくす分、データ円をJPEG圧縮から守る装置が必要となる。本研究ではデータ円それぞれに色付きのボーダーをつけることにした。

4.3.1節では電子透かしの実装方法、4.3.2節ではコードマークサイズの縮小化について述べる。

4.3.1 電子透かしの導入

RICの存在判断のため用いる電子透かしは二つの特徴が必要となる。

- リサイズ・JPEG圧縮に耐えられる強い電子透かし。

- 画像を損傷させない電子透かし。

RIC自体がリサイズとJPEG圧縮のみを対象にしているため、電子透かしもその二つのみに強く対応できれば良い。本研究は上記の特徴をChengら [63]の電子透かし手法を用いて実装した。離散ウェーブレット変換(DWT)を用いる電子透かし手法は、レベル1のDWTの中・低周波数領域はJPEG圧縮によりデータが劣化されやすいため、レベル2やレベル3のDWTが多く採用されている。Chengらの手法もレベル2のDWTを採用し、Figure 4.8のように、HL2領域に離散コサイン変換(DCT)を用いて電子透かしの埋め込む。

それに加え、遺伝的アルゴリズム(Genetic Algorithm, GA)を電子透かしに導入し、画像を最大限損傷させず、画像劣化に最大限ロバスト性をもつ最適なパラメータを探すアルゴリズムを提案した。Figure 4.9で示しているように、画像にあるパラメータで電子透かしの埋め込み、本体の画像が埋め込まれる前の画像との損傷度(PSNR)が一定値以上であり、かつ、画像を劣化させて抽出した電子透かしが本来の電子透かしとの一致率(NCC)が一定値以上になるまで、繰り返すものである。総 n 回のループが回るとして、 $i+1(1 \leq i < n)$ 回目のループは i 回目の情報をもとにパラメータを算出する部分がGAをもとにする部分である。この方式を、画



Figure 4.8: 2-level DWT [63]

像劣化はJPEG 30%の圧縮、PSNRは30以上の閾値を置く実装としてRICに取り組んだ。その結果、本来の画像をなるべく損傷しないJPEG圧縮にロバスト性の高い電子透かしを実現できる。詳細コードを付録Bに述べる。

電子透かし埋め込みは、まず画像を小さい辺が基準サイズになるようリサイズする。その後、画像の真ん中から画像の中をいっぱい埋め込める正方形をとり、そこに電子透かしデータを埋め込む。基準サイズが420pxの場合、正方形として取れるピクセル数は 420^2 の176,400となる。そこから、レベル2のDWTをかけると

対象ブロックのHL2領域はピクセル数が1/16に減り、11,025となる。さらに、DCT変換で電子透かしを埋め込める量は1/64bpp(bit per pixel)であり、埋め込めるデータ量は172ビットとなる。その結果、電子透かしとして埋め込める最大サイズの画像は、正方形の場合、13x13の白黒の画像となる。最初の基準サイズが640pxの場合、同じ計算で20x20の白黒画像を埋め込める。

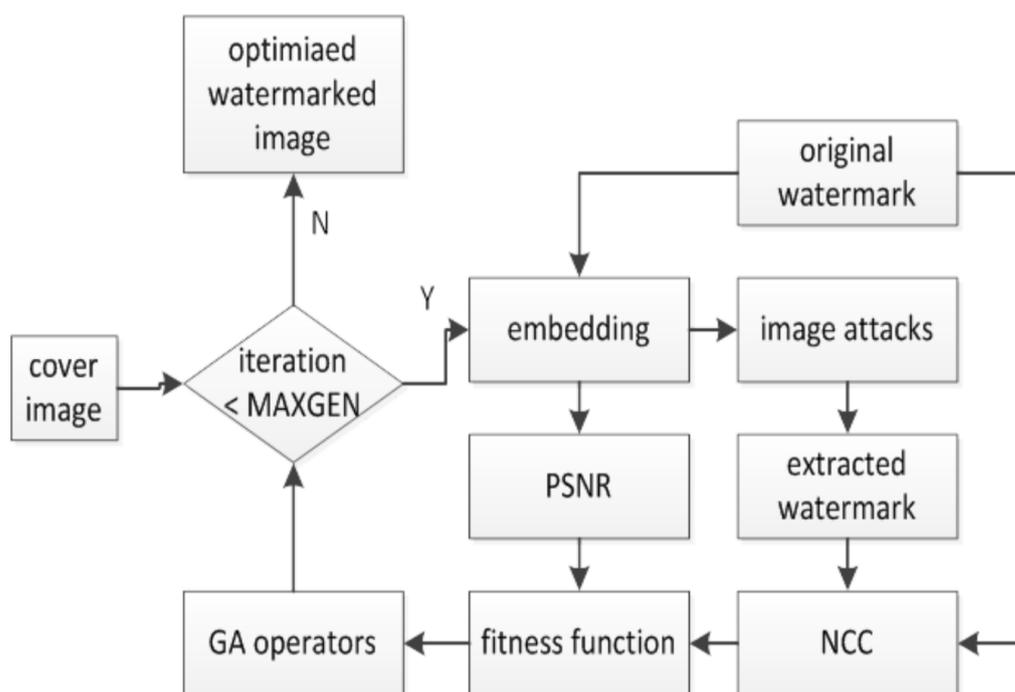


Figure 4.9: GA をもとにした最適な電子透かし探しアルゴリズム [63]

電子透かしの実装により、グレーボックスをなくし、RICに対し様々なデザインが可能になる。Figure 4.10、Figure 4.11に新しいデザイン例2つを表示する。グレーボックスがなくなったため、画

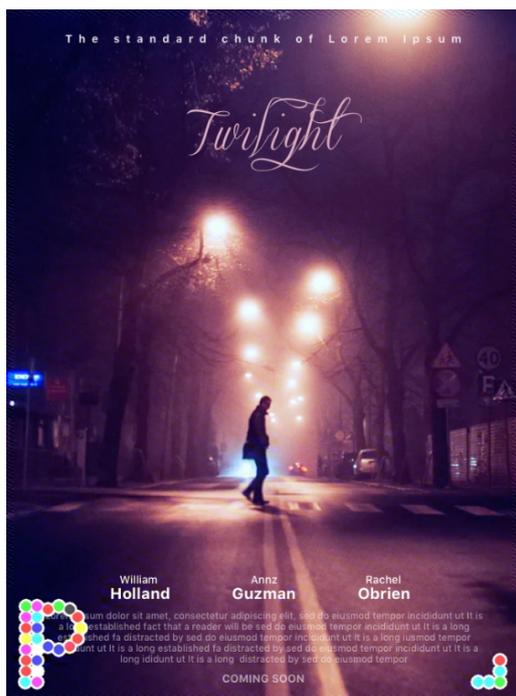


Figure 4.10: RIC デザイン v0.2



Figure 4.11: RIC デザイン v0.3

像のどこにも配置可能になる上、必ずコードマークがくつつく必要のなくなり、様々なデザインが可能になった。しかし、JPEG圧縮に対し、既存のRICと同様のロバスト性を保つためには、Figure 4.3で示したコードマークのサイズより小さくすることができない。従って、コードマークサイズに関する改善も必須となる。

4.3.2 コードマーク縮小化

コードマークのデザイン幅を広げたが、未だにサイズが大きく、画像とバランスが良くない。また、画像の上にコードマークを埋め込んでいるため、著作権問題を解決できていない。そこで、本

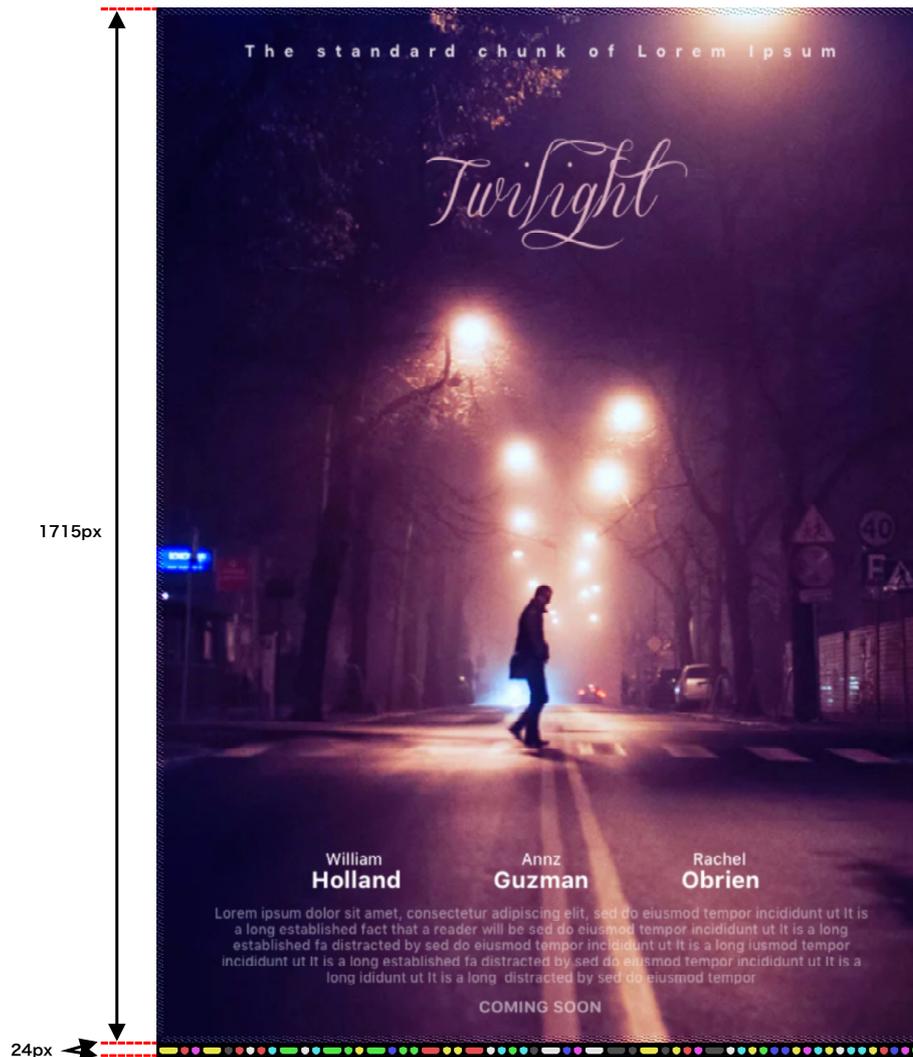


Figure 4.12: RIC デザイン v1.0

研究は、コードマークを画像の外側に埋め込む、及び、コードマークサイズを極端に小さくすることにした。その結果を Figure 4.12 に示す。本体の画像の下部にコードマーク領域を付け足す形で、画像の著作権を侵害しない。また、データ円は一行に並べることで、コードマーク領域の高さを極めて低くした。さらに、データ円の半径サイズを幅 1280px の画像を基準にした時、既存 13px から

7pxに小さくした。Figure 4.12に示す通り、高さ1,739pxのうち、画像部が1,715px、コードマーク部が24pxであり、1.38%の領域のみをコードマークとして使用する。しかし、その分データ円の数は増え、58個の円が存在し、174ビットを埋め込める。

その結果、画像の内容を妨げない上、画像がコンテンツに紐づいていることを下部の一線でユーザーが認識できることにより、RIC v0.1のデザイン問題を解決できた。しかし、RIC v0.1は画像が420px以上であれば、JPEG圧縮・リサイズに関しロバスト性を持っていたが、データ円のサイズが小さくなった分、ロバスト性保つことができない。そのため、本研究では画像の最少サイズを640pxに上げ、JPEG圧縮・リサイズに関しロバスト性を保てるように、RIC読み取りの最適化を行った。

第5章

Image SuperDistribution System

この章では本研究の二つの研究目的のうち、コンテンツ配信モデルに関して述べる。2.1.4節で述べた既存コンテンツ配信モデルに対する問題を解決するにあたり、超流通システム [6] が一番理想の形である。超流通はコンテンツ配信におき理想の配信モデルであり、コンテンツを保護しながら、様々なユーザーにコンテンツを届けられる。しかし、現在、完全な超流通は実現できていない。

超流通を実現し、Amazon、Netflixなどのプラットフォーマーに依存せず、直接Sony、Disneyなどのプロバイダーがコンテンツを配信するには、プラットフォームに依存しない独立した媒体が必要である。現在は、URLと画像のみがプラットフォーム独立型の媒体であり、特別なアプリケーションなしで情報を読める。従って、本研究は4章で提案したRobust Index Code (RIC)を用いて、画像をコンテンツへアクセスできる媒体として選び、新しい配信モ

デルとなる画像超流通システム（Image SuperDistribution System、ISDS）を提案する。

ISDSのアプローチは、プロバイダーがコンテンツをアップロードし、ユーザーが再生・課金するまでの全体のフローをプラットフォームに依存せず、直接構成されるモデルである。また、デジタル画像にRICを用いてコンテンツを紐付けて、画像のみでコンテンツを配信できる部分もアプローチである。

5.1節ではビジネスモデルの概要の説明、5.2節ではビジネスモデルの例の紹介、5.3節ではISDSの概要、5.4節ではデジタル画像を媒体として利用する際の欠点、5.5節ではISDSのSDKの配布戦略、5.6節ではISDSの構造、5.7節ではISDSのアプリケーション図、5.8節ではコンテンツ配信プロセスについて述べる。

5.1 ビジネスモデルとは？

本研究で提案するISDSはコンテンツ配信に関するビジネスモデルイノベーション研究に属する。そのため、ISDSの概要を述べる前に、ビジネスモデルの概念を先に述べる。Afuahはビジネスモデルは「儲けるための仕組み」と定義している [64, 65]。しかし、ビジネスモデルにはさらに広い概念であると指摘され、谷井らはビジネスモデルを以下のポイントで定義する [65, 66]。

1. 顧客価値の提供
2. 利益方程式(顧客への価値提供方法を財務的側面から明確化)
3. カギとなる経営資源
4. カギとなるプロセス

これらのポイントを要約すると、谷井らは「ビジネスモデルは経営資源を価値に変換して顧客に提供し、自社に利益を創出する仕組み(一連のプロセス)」[65]と定義している。デジタルコンテンツ配信分野においては、月額制の見放題ビジネスモデルが例として挙げられる。そのモデルは、経営資源となる複数のコンテンツプロバイダーから集めたコンテンツを顧客に提供し、月額の決済により自社利益を獲得する仕組みと言える。また、インターネット広告分野ではGoogle AdSenseが例として挙げられ、経営資源となる様々な広告主から集めた広告を、顧客となるインターネットサイトの主(Affiliator)に簡単に提供、その広告収益の一部を自社利益として獲得するモデルだと言える。しかし、ビジネスモデルが機能していても、そこには問題点があったり、全ての要件を満たせるモデルがないため、ビジネスモデルにもイノベーションが必要になる。マークらは以下の場合に、ビジネスモデルイノベーションが必要となると述べている [66]。

1. 既存のソリューションが高すぎる、あるいは複雑すぎるため、市場から完全に排斥されている膨大な潜在顧客が抱えているニーズに破壊的イノベーションで対応するチャンス。
2. 目新しい技術を新しいビジネスモデルによって提供するチャンス。
3. 「解決すべきジョブ」という視点が存在しない分野に、この視点を持ち込むチャンス。
4. 価格破壊者に対抗する必要性。
5. 競争基盤の転換に対応する必要性。

このポイントより、ビジネスモデルイノベーションは新たな環境を作り出し、企業に大きなビジネス機会を与えられる新たなビジネスモデルと言える。本研究では、デジタルコンテンツ配信分野に関しては、上記のビジネスモデルイノベーションが必要となる場合の中、1と2の部分により、イノベーションが必要だと認識し、新たな配信インフラを提案している。膨大な潜在顧客が存在すると予想しているトップティアコンテンツ配信に関し、コンテンツプロバイダーが抱えているニーズを解決でき、今までモバイル環境においてアプリケーションなしで不可能だったトップティアコンテンツ配信を、デジタル画像を用いる新しい技術RICによ

り可能にすることが本研究の目的であり、新たなビジネスモデルを構築できる環境を与えることにより、本研究のISDSがビジネスイノベーション研究である理由だと言える。

5.2 ビジネスモデルの例

この節では、5.1節で定義したビジネスモデルの例を紹介する。5.2.1節では実用化しているビジネスモデルの一つである、Googleが提供している広告配信インフラであるGoogle AdSense [67]、5.2.2節ではビジネスモデルの研究の一つである、著作権保護における電子透かしサービスモデル [68]を紹介する。

5.2.1 Google AdSense

Google AdSenseは2003年からGoogleが提供するコンテンツ連動型広告配信サービスである。その仕組みをFigure 5.1に示す。広告主はAdSenseに広告を登録するだけで、広告がインターネットに勝手に配信される。各インターネットサイトのオーナーであるアフィリエイトは、Google AdSenseに広告枠を登録し、Googleが提供するHTMLタグを自分のサイトに貼り付けるだけで、広告を配信できる。ユーザーが広告を視聴又はクリックにより、収益が発生し、アフィリエイトに戻ってくる。従って、アフィリエイト

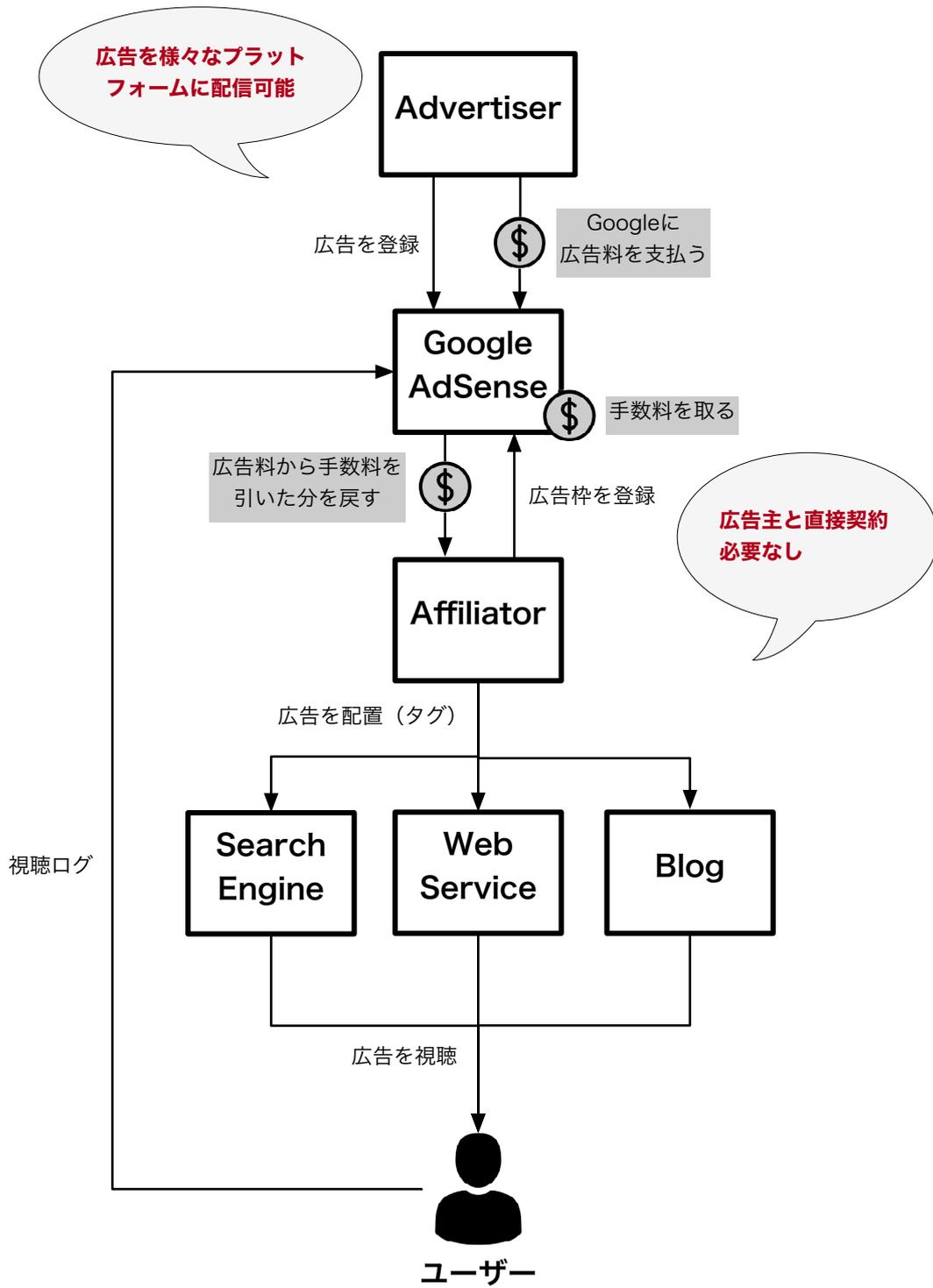


Figure 5.1: Google AdSense の仕組み

ターは簡単に1ヶ月で数千ドルやそれ以上の収益を得ている [69]。技術的な優秀さよりも広告配信の簡単さゆえに、たくさんのアフィリエイトが使用するが、その分他の企業でもサービスを作ることができ、Amazonアソシエイトのような同じ機能をするサービスも活性化している。経営資源となる広告を、簡単なプロセスで配信可能にすることで、ビジネスイノベーションを起こしたと言える。本研究は経営資源となるトップティアコンテンツを簡単なプロセスでモバイル環境上で配信することを目的とし、コンテンツ業界でビジネスイノベーションを目指している。

5.2.2 著作権保護における電子透かし

平川 [68] はコンテンツ著作権侵害の問題点を解決するため、電子透かし技術を用いた著作権保護ソリューションを提案する。平川は、既存に存在する電子透かし技術を用い、人手を介せず、自動で画像をダウンロードするユーザーごとにIDを付け、そのID情報を電子透かしとして画像に埋め込むシステムを構築し、画像を販売している実際のEコマースサイトとPOC（Proof of Concept、機能検証）を行なっている。その結果、不正利用コンテンツの前月比60%減を達成している。平川が提案しているモデルは、Google AdSenseと同じく既存技術を用いて問題を解決している。又、コ

コンテンツの管理が全自動で行うポイントが、AdSenseでアフィリエイトがHTMLタグだけ入力すると全自動で広告が配信されるポイントと同様に簡単なプロセスを主張している。本研究はここまで述べてきた二つの例と同様、コンテンツを簡単なプロセスで配信するビジネスモデルの実現を目的としているが、新しいビジネスモデルによって新技術であるRICを提供するところが、異なる点と言える。

5.3 Image SuperDistribution Systemの概要

既存コンテンツ配信インフラは、プロバイダーがプラットフォームにコンテンツを委託し、プラットフォームがそのコンテンツを自分のアプリ上で配信する形を取っている。2.1.2節で述べた通り、モバイル環境上で暗号化したコンテンツを再生及び保存するには、ネイティブアプリケーションが必要になり、プラットフォームがインフラ上で主役になりつつある。プロバイダーは自らコンテンツを直接配信したいが、プラットフォームを別途で開発・運用する以外に直接配信できるインフラが存在しない。本研究で提案するISDSは、既存ビジネス上ではなかったコンテンツ直販ルートを開拓し、膨大な潜在ユーザーが抱えるニーズのあっ

たトップティアコンテンツ（人気作・最新作）を容易に配信させ、破壊的イノベーションを起こすビジネスモデルイノベーションである。

本研究で提案するビジネスモデルには4つの要件が存在する。

- (1) 簡単に共有できるコンテンツ媒体
- (2) 動画、電子書籍、音楽など、ほとんどのデジタルコンテンツを安全に配信できる
- (3) インターネットの共有の過程で起こる劣化に対し、高いロバスト性をもつ
- (4) プロバイダーがコンテンツのポリシーを決定

要件(1)を解決するため、プラットフォームに依存しないメディアをコンテンツの媒体として選ぶ必要がある。現在URLと画像のみが独立したメディアである。URLに比べて、画像は訴求力のあるメディアであり、ユーザーに画像の中身に興味を持たせる力がある。そのため、画像が要件(1)を解決できる唯一の媒体と判断し、全体システムの名前をImage SuperDistribution System (ISDS)と名付けた。

要件(2)、(3)は、4章で提案したRICを用いる。RICはロバストなカラーコードマークであり、デジタル画像にコンテンツのイン

デックスを埋め込むことで、画像とコンテンツを紐付けられる。コンテンツ自体は画像とは別で存在するため、安全に守ることができ、コンテンツに問題が発生した場合は画像との紐付けを切ることでユーザーからの接続を遮断でき、RICは画像劣化に強いロバスト性をもつため、要件(2)、(3)を解決できる。

最後に、コンテンツをリモートデータベースに保存する際、コンテンツに利用ポリシーや値段を一緒に保存し、ユーザーのアクセスを全てチェックすることにより、プロバイダーがコンテンツのポリシーをリアルタイムでコントロールできる。また、モバイルネイティブアプリケーションのガイドライン [15, 16] により、デジタルコンテンツに関する全てのトランザクションはAppleとGoogle Playのアプリ内決済システムを利用しなければならない。そこで、デバイスの固有IDとトランザクション情報を一緒にリモートデータベースに保存することにより、本研究のコンテンツ配信モデルで別途に決済システムを構築する必要がなく、AppleとGoogleのセキュアな決済モジュールを会員登録なしで指紋認証や顔認証で簡単に利用できる。従って、ISDSはすべての要件を満たし、実現可能なモデルとなる。

5.4 デジタル画像の欠点

ISDSはデジタル画像にコンテンツを紐付け、コンテンツを配信するインフラモデルである。画像は訴求力があり、コンテンツ配信に最も適した媒体だと言える。しかし、モバイル環境で画像を配信することは容易でない。Facebookなどの既存アプリケーションからRIC SDKに画像を直接読み取るように指示することができない。従って、画像を一回ギャラリーに保存し、ギャラリーアプリから画像を開く必要がある。

そのため、本研究は画像のみではなく、URLも配信媒体として用いる。主な利用ケースは下記の通り2パターン存在すると見込む。

1. 初回コンテンツへアクセス：既存アプリケーションのURLをクリックし、コンテンツへアクセスする。コンテンツ再生画面でコンテンツにアクセスできるRIC画像をギャラリーに保存する。
2. コンテンツへの再アクセス：ギャラリーアプリからコンテンツの画像を探し、開くことでコンテンツへ再アクセスできる。

ユーザーは最初URLをクリックする1ステップでコンテンツへアクセスでき、コンテンツにアクセスできる画像を保存すること

により、既存プラットフォームに依存せずコンテンツへのアクセス・リテンションが可能になる。URLは初回のアクセスのみで利用し、保存プロセスでは用いないため、URLの文字を間違えるなどの問題が生じない。従って、デジタル画像の欠点を克服できる。

5.5 SDKの配布

コンテンツと紐付けられたRIC画像を読み取るには、RIC SDKが必要である。RIC SDKは主にRIC画像からコンテンツのインデックスを抽出し、インデックス先のコンテンツを再生する役割を行うネイティブアプリケーションのライブラリである（iOSのFramework、Androidのaar）。

RIC SDKは以下の大きく3つのレイヤー層を持つ。

RICモジュール

RICの読取機能を提供するソフトウェアモジュールである。RICモジュールはISDSのみならず、第三者が利用できるよう、汎用的なAPIを提供する。ただし、RICのパラメータに対してセキュリティの担保が必須不可欠な為、RIC読取部分はC++で実装、コンパイルされたバイナリーがSDKに組み込む仕様とする。

クローズドモジュール

RIC モジュールと同様、セキュリティの担保が必要な部分を事前コンパイルし、バイナリーとしてSDKに組み込むモジュールである。主にDRM再生、コンテンツサーバーとの通信、決済などがこのモジュール内で実装し、汎用的なAPIとして提供する。

オープンモジュール

コンテンツ再生プレイヤーのユーザーインタフェースなど、第三者がカスタマイズ出来るようにしたモジュールである。このモジュールはオープンソースとして公開し、モジュール自体のカスタマイズを許す。さらに、サーバーとの接続プロトコルの設定もオープンモジュールに位置している為、ISDSが中継するインフラが中断しても、プロバイダーは自らSDKを直接繋げることを可能にしている。

上記の機能を持つRIC SDKは、事前にユーザーのデバイスにRIC SDKが組み込まなければいけない。コードマークや電子透かしのSDKは普通モバイルアプリに組み込まれるSDKとして実装されるケースが多いが [70]、ユーザーにとってSDKのインストールは容易でない。それではすべてのユーザーがSDKを持つようにするにはどうすれば良いだろうか。

無論、RIC SDKをISDSに興味を持つ個々のアプリやWhatsApp、Facebook、Twitterのように最もダウンロードされたアプリ [71, 72] にSDKを搭載してもらう方法が存在する。しかし、本研究で提案するモデルはユーザーの追加アクションなしでRIC SDKをユーザーのデバイスに埋め込むことである。従って、本研究はApple、Google、Huaweiのようなスマートフォン製作側又はAT&T、ドコモ、Orangeのような通信キャリア側とコンタクトし、デバイスの出荷段階でRIC SDKを搭載してもらうことにより、新たなアプリケーションのインストールなしですべてのユーザーがRIC画像を読み取れるインフラを構築する。SDKを搭載する側のメリットは、簡単にトップティアコンテンツを自社プラットフォームで再生可能になり、ユーザーと広告案件が増えることである。

5.6 Image SuperDistribution Systemの構造

本節では、ここまで述べてきたISDSの構成要素を用い、トップティアコンテンツをどう配信するか、その全体構造について述べる。先に、既存コンテンツ配信インフラである、プラットフォームを経由した配信構造をFigure 5.2に示す。

まず、プロバイダー（CP）は、MG（Minimum Guarantee）とR/S

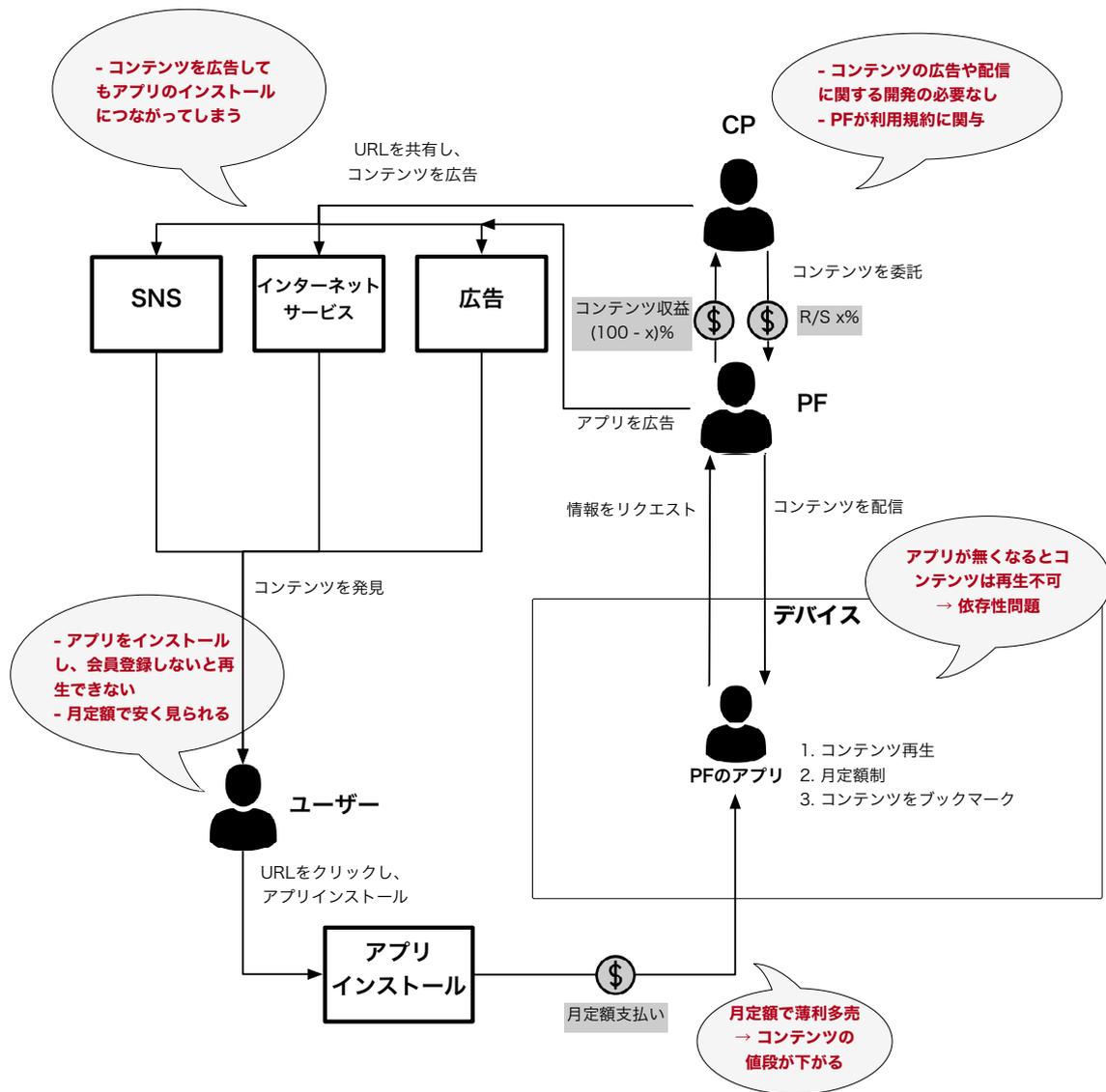


Figure 5.2: コンテンツプラットフォームの配信構造図

(Revenue Share) の契約のもと、コンテンツをプラットフォーム (PF) に委託する。プラットフォームはコンテンツを自社のアプリ、自社のサーバーを通じて配信を行う。また、プラットフォームはアプリ自体をマーケティングするため、プロバイダーは別途コンテンツの広告を行う必要がない。その上、ユーザーは一つ

のアプリ上で様々なコンテンツを見放題で再生できる利点が存在する。

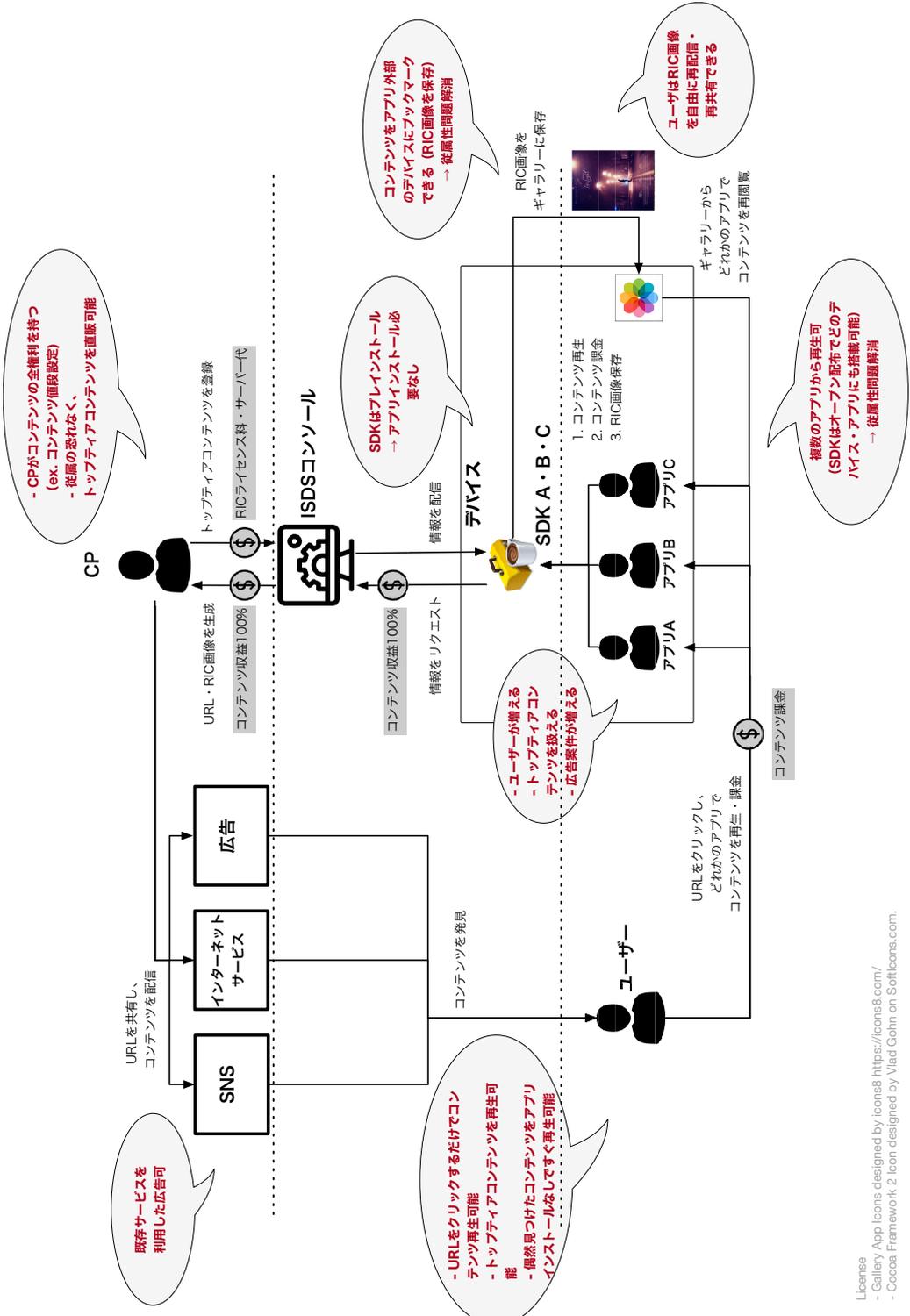
しかし、この構造には短所も存在する。プロバイダー側にとっては、自らコンテンツの広告を配信することに制限がある。なぜなら、ユーザーは最終的にプラットフォームのアプリをインストールする画面に遷移されるためである。また、プラットフォームアプリへユーザーが集まり、コンテンツがそのアプリの中で再生・保存されるため、プラットフォームへの従属性問題が生じる。プラットフォームがコンテンツの利用規約に関与したり、コンテンツキュレーションについて強く主張できたり、月額サービスにより薄利多売現象が発生することがある。さらに、トップティアコンテンツとなる、2018年平均視聴率のランキング10位までのドラマ [73] の月額制動画配信サービスのプロバイダーへの配信状況を調査して結果、TABLE 5.1に示す通り、プロバイダーはほとんどのトップティアコンテンツをプロバイダーに委託せず、その原因は従属性問題にあると推測され、その結果、ユーザーは人気作を楽しめない。

今度は、本研究で提案するISDSの配信構造をFigure 5.3に示す。ISDSはプロバイダー（CP）が直接コンテンツを配信するモデルなため、プロバイダーが様々な部分を自由にコントロールできる。

ドラマ	放送局	視聴率	プラットフォーム			
			Netflix	Amazon Prime Video	U-NEXT	dTV
99.9 -刑事専門弁護士- SEASON 2	TBS	17.4%	X	X	X	X
リーガルV ～元弁護士・小鳥遊翔子～	テレビ朝日	15.6%	X	O	O	O
相棒 season17	テレビ朝日	15.3%	X	X	X	X
BG ～身辺警護人～	テレビ朝日	15.1%	X	X	X	X
ブラックペアン	TBS	14.3%	X	X	X	X
義母と娘のブルース	TBS	14.1%	X	X	X	X
特捜9	テレビ朝日	13.9%	X	X	X	X
下町ロケット	TBS	13.2%	X	X	X	X
未解決の女 警視庁文書捜査官	テレビ朝日	12.8%	X	X	O	X
木曜ミステリー 科捜研の女 season18	テレビ朝日	12.5%	X	X	X	X

TABLE 5.1: 2018年平均視聴率のランキング上位ドラマの月額制動画サービスへの配信可否調査結果

まず、プロバイダーはコンテンツとその利用規約をISDSコンソールに登録し、URLとRIC画像を生成する。次に、生成したURLを用い、既存インターネットサービスに広告として配信する。コンテンツ再生に必要なRIC SDKは、5.5節の戦略により、ユーザーのデバイスに事前に搭載しているため、プロバイダー側で別途必要な作業はない。従って、ユーザーはインターネットからコンテン



License
 - Gallery App Icons designed by icons8 <https://icons8.com/>
 - Cocoa Framework, 2 Icon designed by Vlad Gohn on [SoftIcons.com](https://www.softicons.com/).

Figure 5.3: ISDS 配信構造図

ツURLをクリックするだけでデバイスのSDKを通じて、利用規約（広告付きの無料再生又は有料個別課金など）の元にコンテンツを再生できる。また、コンテンツはRIC画像としてデバイスのギャラリーに保存され、ギャラリーから再閲覧が可能となる。その結果、コンテンツを再生するために必要なアプリはSDKを搭載していれば、どのアプリでも再生でき、一つのアプリケーションに依存しない環境が構築できる。

収益の面は、SDKを搭載した側で発生したコンテンツ収益がISDSに一回集まり、各プロバイダーに還元する。その過程でISDSの運営に必要なサーバー代、RICのライセンス料をプロバイダーに請求により、ISDS運営側とプロバイダー側の両側に収益が発生する仕組みとなる。

しかし、ISDSにも短所が存在し、既存ではプラットフォーマーが代行で行っていた、コンテンツのマーケティングやユーザー集めをプロバイダーが直接行う必要がある。配信に関する自由度が高まり、トップティアコンテンツを配信できる一方、営業に関してもプロバイダーが直接行う必要がある。

5.7 アプリケーション構築

Figure 5.2に示した既存のコンテンツ配信構造に対し、プラッ

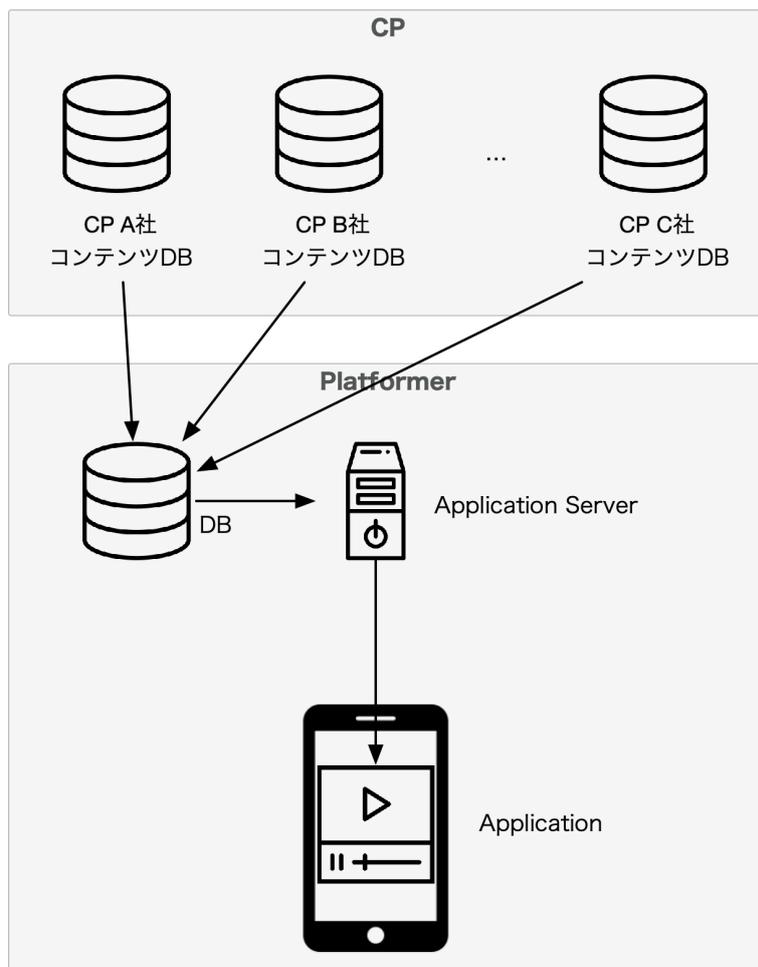


Figure 5.4: コンテンツプラットフォームのアプリケーション図

プラットフォームのアプリケーション図を Figure 5.4 に示す。プロバイダーはコンテンツをプラットフォームに委託し、プラットフォームが全てのコンテンツを自社のデータベースに登録を行う。ユーザーはプラットフォームのアプリケーションをインストールし、プラットフォームの Application Server を通じてコンテンツを閲覧する。全ての配信に関する動作が、プラットフォームの環境内で収まる。プロバイダーはコンテンツ配信に関して

必要な作業がないということが長所になるが、プラットフォームのアプリケーションなしではコンテンツ配信が不可能になり、従属性が生じる短所がある。それに対する、Figure 5.3に示した本研究で提案するISDSの配信構造に対する、インフラのアプリケーション図をFigure 5.5に示す。

ISDSにおいては、既存プラットフォームモデルに加え、ISDSのコンソール (CMS、Content Management System)、Application Server、リーダーのデバイス・アプリ側を追加した。ISDSレイヤーは[74]と同様、DB、サーバーの3層 C/Sシステムとなる。コンテンツ情報はプロバイダーがコンソールに登録すると、DBにタイトルなどのコンテンツメタデータ、コンテンツ利用規約、コンテンツ再生に必要なCDN情報などが格納される。その過程で生成されたインデックスをポスター画像にRICを用いて埋め込み、生成したRIC画像とURLをプロバイダーに提供する。アプリケーションサーバーはSDKからリクエストされるコンテンツ情報を送信し、CDN情報によりSDK内部のプレイヤーでコンテンツが再生される。ISDSコンソールはAngular Framework [75]で実装し、アプリケーションの画面を付録Cに示す。アプリケーションサーバーはSpring Framework 4.3.5 [76]とTomcat 8.5 [77]を用い、DBはMySQL 5.7 [78]を用いた。

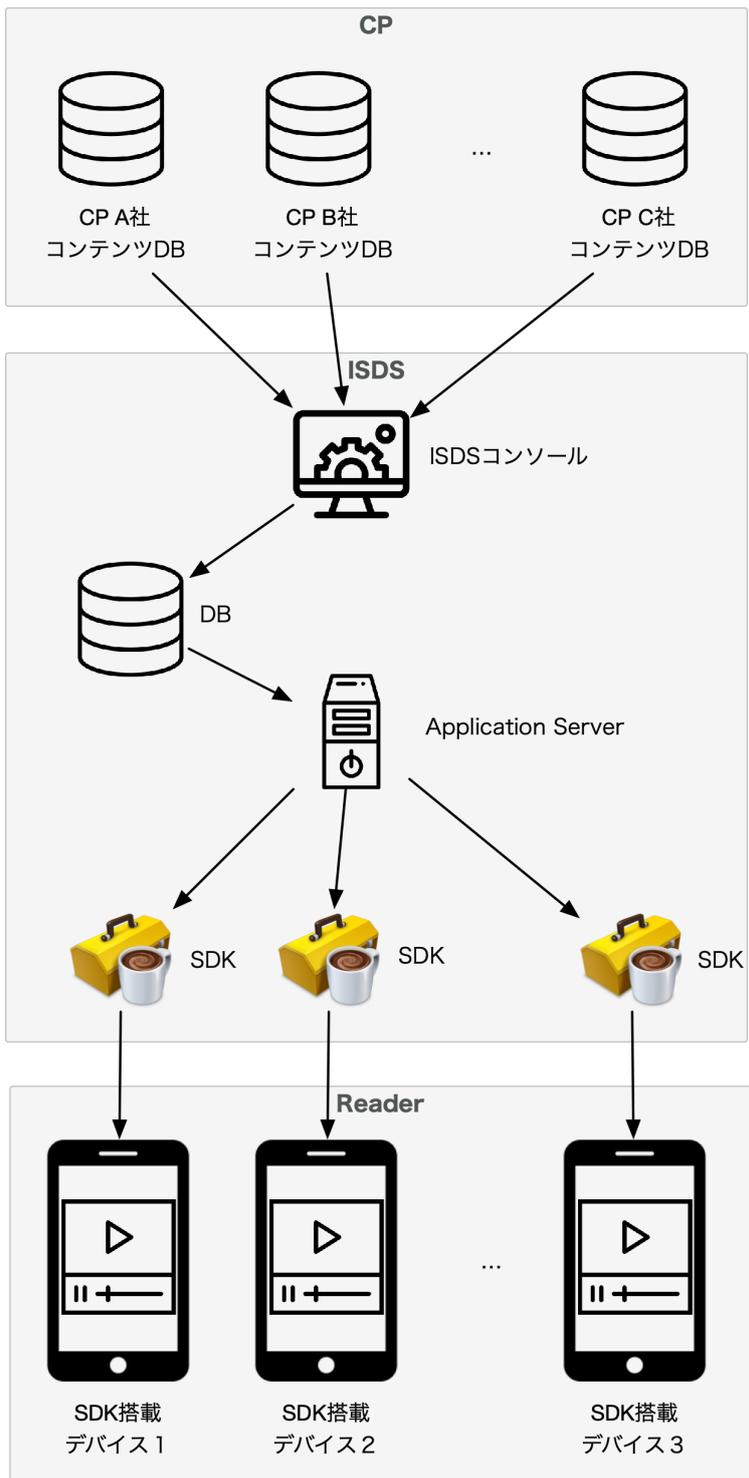


Figure 5.5: ISDS のアプリケーション図

Content Provider side

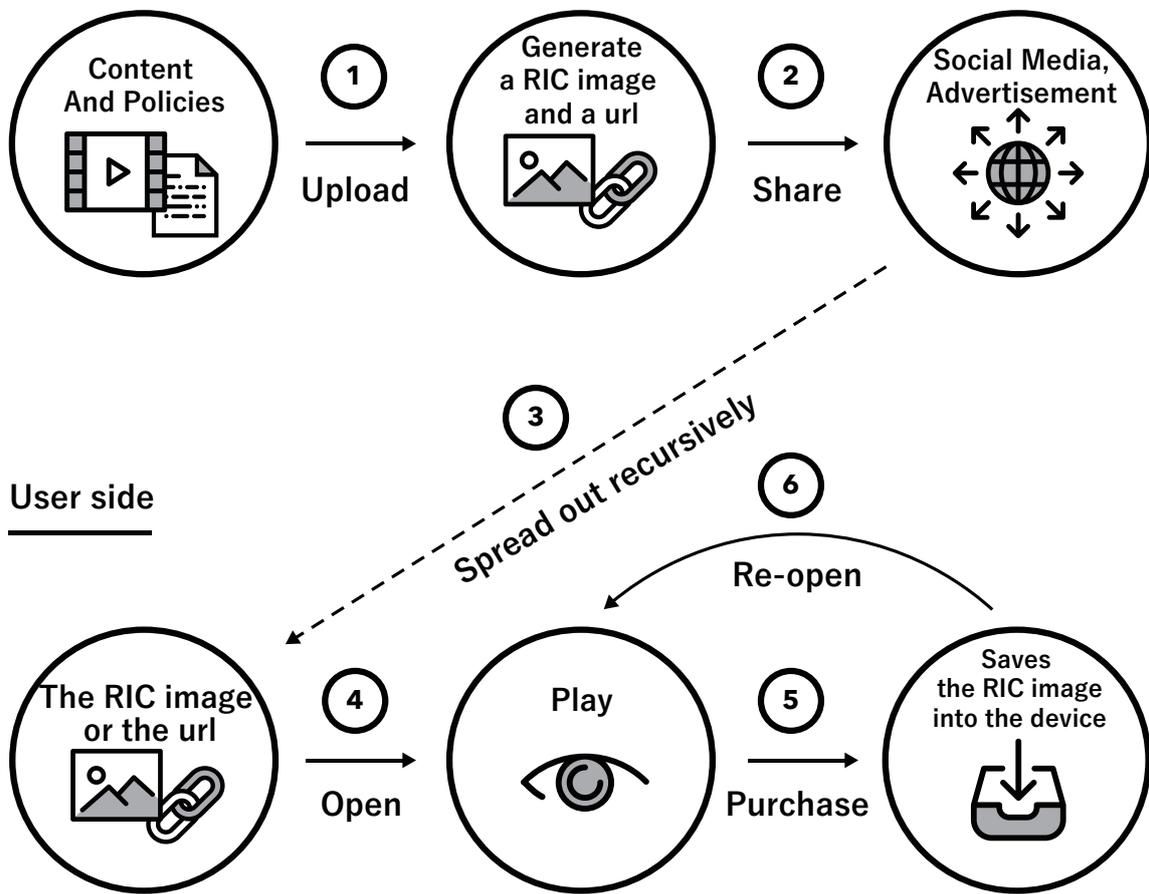


Figure 5.6: ISDS におけるコンテンツ配信プロセス図 [2]

5.8 コンテンツ配信プロセス

既存のコンテンツ配信プロセスに比べ、ISDS と一番大きな違いは、プラットフォームに依存せず、コンテンツを簡単に配信・消

費できることである。又、ユーザーはプロバイダーが決めた利用ポリシーに従いコンテンツを再生する。

5.5節で述べた通り RIC SDK をほとんどのユーザーが持っているとして仮定すると、ISDS は Figure 5.6 に示すコンテンツ配信プロセスによって実現する。プロバイダーは 2 ステップでコンテンツを配信でき、ユーザーは 3 ステップでコンテンツを再生・課金できる。5.8.1 節でプロバイダー側を、5.8.2 節でユーザー側のプロセスに関して述べる。

5.8.1 コンテンツプロバイダー側

ISDS 上でプロバイダーがコンテンツを配信するには、コンテンツをリモートに置き、画像と紐付ける必要がある。本研究では ISDS コンソール（コンテンツプールコンソール）を実装し、配信のプロセスを簡略化した。プロバイダーがコンテンツとポスター画像をコンソールにアップロードすると、システムがコンテンツをデータベースに保存し、その過程で生成されたインデックスをポスター画像に RIC を用いて埋め込む。出来上がった RIC が埋め込まれたポスター画像と URL をプロバイダーに提供する。プロバイダーは URL や画像をインターネットで共有するだけで、コンテンツ配信が完了する。共有された URL や画像はインターネット上

のどこからもアクセスでき、SNSや広告で数回コピーされることで簡単に広がっていく。5.4節で述べた通り、画像を既存アプリ上で直接開くことは困難なため、この段階ではURLを主に用いるべきと言える。

5.8.2 ユーザー側

コンテンツ広告やSNSなどで拡散により、URLやRIC画像がユーザーに露出される。ユーザーはそれらをRIC SDKを通じて（画像やURLからインデックスを抽出）コンテンツを再生できる。SDKはデバイスに事前設置しているため、コンテンツ再生はWebではなくネイティブに実行し、既存プラットフォームと同様のセキュリティが担保できる。課金決済もAppleやGoogleのトランザクションを用いるため、セキュアな決済ができる。また、コンテンツ自体はリモートデータベースに存在し、ストリーミングされるため、ユーザーはコンテンツにアクセスできる、デバイスの容量を極力使用しないデジタル画像としてギャラリーに保存できる。ユーザーは一回RIC画像をギャラリーに保存すると、紐づいているコンテンツはギャラリーからまたいつでも再生できる。さらに、保存した画像を削除しても、コンテンツはリモート上に残っているため、他のRIC画像から再アクセスが可能になる。そ

の上、RIC 画像はリサイズや JPEG 圧縮に対してロバスト性を持つ為、ユーザーは他のユーザーへ URL 又は RIC 画像を共有するだけで 2 次拡散できる。要約すると、ユーザーは URL や RIC 画像を開く 1 ステップでコンテンツを再生でき、課金に 1 ステップ、画像保存に 1 ステップ、共有・2 次拡散に 1 ステップで簡単にコンテンツを消費できる。

第6章

予備実験

本研究は新たなコンテンツ配信モデルである ISDS を提案し、その構成要素である RIC を提案した。本章では、RIC v0.1 の性能実験、ISDS に特化した RIC v1.0 の性能実験を行う。

RIC v0.1 は画像のサイズが 420 ピクセル以上、JPEG 圧縮が 30%以上であれば、100%データの抽出に成功した。また、ISDS 用に特化した RIC v1.0 は、コードマークサイズを小さくすることにより、画像サイズが 640 ピクセル以上、ピーク信号対雑音比 (Peak Signal-to-Noise Ratio、PSNR) が 16 以上であれば、100%データの抽出に成功した。

6.1 Robust Index Codeの予備実験

6.1.1 実験概要

実験は4つの実験を行う。RIC v0.1のロバスト性実験、RIC v1.0の電子透かしロバスト性実験、RIC v1.0のロバスト性実験、関連研究との比較実験を行う。RICのロバスト性実験は、RICが画像劣化に対し、100%データが抽出できることを確認する。電子透かしは、RICの最小のサイズを決めるための実験であり、実験から画像の基準サイズを決定し、画像劣化のロバスト性実験でそのサイズ以上の画像劣化に対し、100%データが抽出できるかを確認する。比較実験は、既存研究であるQR Code [18]が画像のリサイズとJPEG圧縮に対し、ロバスト性が低いことを証明する。

実験に使用するRIC v0.1のパターンは、Figure 4.2のパターンを使用する。そのパターンのパラメータは r を4、 d を2、 D_0 を80、 D_1 を255、 p が6、 N_c が22、 N_d が 2^{12} となる。RIC v1.0のパターンは r を2、 d を2、 D_0 を80、 D_1 を255、 N_c が58、 N_d が 2^{20} となる。実験に使用する画像は10年間撮ってきたプライベートな画像9,850枚と、画像サイト [79]からダウンロードした150枚を用い、幅が1280ピクセルである10,000枚を使用する。また、本来の画像と劣化された画像の差はピーク信号対雑音比 (Peak Signal-to-Noise Ratio、



Figure 6.1: リサイズ 420px × JPEG 30% 圧縮を適用した RIC v0.1 [12, 14]

PSNR) で表現する。PSNR の値が大きいほど、本来画像に近いことを意味する。PSNR が無限大の時、本来画像と同じであることを意味する [21, 80]。データ検証に必要なデータベースは、本研究の実験ではメモリ上で保存することにする。

6.1.2 RIC v0.1 のロバスト性実験

RIC のロバスト性は本研究で対象にしているインターネット上で最も起きる画像劣化、リサイズ、JPEG 圧縮 [41] に対して実験を行う。また、2 つの画像劣化を複合した劣化も対象とする。TABLE 6.1 に実験で適用する 16 個の画像劣化を示す。画像の劣化の具合を例えるため、Figure 6.1 に劣化タイプ 9 番の RIC を示す。

TABLE 6.2 の実験結果により、リサイズ 420 ピクセル、JPEG 30% 圧縮以上の画像劣化に対し、100% データの抽出に成功した。

番号	劣化タイプ
1	なし
2	JPEG 50% 圧縮
3	JPEG 30% 圧縮
4	リサイズ 2000px
5	リサイズ 2000px × JPEG 50% 圧縮
6	リサイズ 2000px × JPEG 30% 圧縮
7	リサイズ 420px
8	リサイズ 420px × JPEG 50% 圧縮
9	リサイズ 420px × JPEG 30% 圧縮
10	リサイズ 420px × JPEG 30% 圧縮 × リサイズ 1280px × JPEG 30% 圧縮
11	リサイズ 420px × JPEG 30% 圧縮 × リサイズ 1280px × JPEG 30% 圧縮 × リサイズ 450px
12	リサイズ 420px × JPEG 30% 圧縮 × リサイズ 1280px × JPEG 30% 圧縮 × リサイズ 450px × JPEG 30% 圧縮
13	リサイズ 420px × JPEG 30% 圧縮 × リサイズ 1280px × JPEG 30% 圧縮 × リサイズ 450px × JPEG 30% 圧縮 × リサイズ 2000px
14	リサイズ 420px × JPEG 30% 圧縮 × リサイズ 1280px × JPEG 30% 圧縮 × リサイズ 450px × JPEG 30% 圧縮 × リサイズ 2000px × JPEG 30% 圧縮
15	リサイズ 420px × JPEG 30% 圧縮 × リサイズ 1280px × JPEG 30% 圧縮 × リサイズ 450px × JPEG 30% 圧縮 × リサイズ 2000px × JPEG 30% 圧縮 × リサイズ 400px
16	リサイズ 420px × JPEG 30% 圧縮 × リサイズ 1280px × JPEG 30% 圧縮 × リサイズ 450px × JPEG 30% 圧縮 × リサイズ 2000px × JPEG 30% 圧縮 × リサイズ 400px × リサイズ 420px

TABLE 6.1: ロバスト性実験劣化タイプ [12, 14]

6.1.3 電子透かしロバスト性実験

本節では Cheng らの電子透かし手法 [63] をもとに実装した DWT 電子透かしのロバスト性に対し実験を行う。電子透かしは画像の

番号	RIC サイズ (px)	平均 PSNR	成功率	データ候補数	読み取り時間
1	204×205	∞	1.0000	1.0000	0.0630s
2	204×205	33.76	1.0000	1.0000	0.0562s
3	204×205	32.22	1.0000	1.0000	0.0572s
4	313×314	42.33	1.0000	1.0000	0.0810s
5	313×314	37.58	1.0000	1.0000	0.0781s
6	313×314	35.81	1.0000	1.0000	0.0715s
7	73×74	31.74	1.0000	1.0000	0.0176s
8	73×74	29.47	1.0000	1.0000	0.0649s
9	73×74	28.64	1.0000	1.0000	0.4000s
10	204×205	27.71	1.0000	1.0000	0.4200s
11	77×78	27.50	1.0000	1.0000	0.4301s
12	77×78	27.47	1.0000	1.0000	0.4152s
13	313×314	27.22	1.0000	1.0000	0.4872s
14	313×314	27.10	1.0000	1.0000	0.4565s
15	70×71	28.02	0.9999	1.0000	0.4210s
16	73×74	27.90	0.9998	1.0000	0.7228s

TABLE 6.2: ROBUST 性実験結果

大きさに比例してロバスト性が増加すると予想でき、本実験では、電子透かしを埋め込む基準画像サイズを RIC v0.1 の最小サイズとなる 420 ピクセルから実験を行う。また、電子透かしはレベル 1 の DWT (LL1_DWT) とレベル 2 の DWT (LL2_DWT) に対しても実験を行う。

本実験で用いられる画像は 2 種類存在し、一つは電子透かしを埋め込んでいない普通の画像（一般画像と呼ぶ）、もう一つは電子透かしを埋め込んだ画像（RIC 画像と呼ぶ）である。理想の結果

は、一般画像は電子透かしが検出されてはいけなく、RIC画像は全て電子透かしが検出されなくてはならない。実験は以下の手順で行い、実験結果をTABLE 6.3に示す。本実験では一致率の基準を0.65（グレースケールの画像は0.58）において実験を行った。

手順a 画像10,204枚をコピーする。

手順b コピーした画像に電子透かしを埋め込む。

手順c 電子透かしを埋め込んだ画像をJPEG 30%の圧縮をかける。

手順d 電子透かしを埋め込んだ画像と、そうでない画像から電子透かしを検出する。

TABLE 6.3の420pxのRIC画像の実験結果より、小さい画像サイズで電子透かしを埋め込むと、JPEG劣化に対してロバスト性が低いことを示す。また、レベル1のDWTよりレベル2のDWTの方がJPEG劣化に対してロバスト性が高い。そのため、RIC画像全てを検出OKの結果が出たLL2_DWT（640px）がRICで適した手法であることを示している。

6.1.4 RIC v1.0のロバスト性実験

6.1.3節の結果により、RIC v1.0は最小画像サイズを640pxに上げなければいけない。そのため、本節のRICロバスト実験では、

メソッド	画像種類	#検出 OK	#検出 NG	平均NC (一致率)	最大NC (一致率)
LL1_DWT (420px)	一般画像	16	10188	0.5032	0.6722
	RIC 画像	2391	7813	0.5727	0.9571
LL2_DWT (420px)	一般画像	505	9699	0.5282	0.6929
	RIC 画像	10197	7	0.9925	1.0000
LL1_DWT (512px)	一般画像	17	10187	0.4975	0.6592
	RIC 画像	9467	737	0.8207	0.9954
LL2_DWT (512px)	一般画像	114	10091	0.5133	0.6532
	RIC 画像	10193	11	0.8950	1.0000
LL1_DWT (608px)	一般画像	14	10190	0.4971	0.6518
	RIC 画像	9438	766	0.8299	0.9967
LL2_DWT (608px)	一般画像	70	10134	0.5153	0.6810
	RIC 画像	10192	12	0.8805	0.9865
LL1_DWT (640px)	一般画像	16	10188	0.4994	0.6446
	RIC 画像	9684	520	0.9096	0.9937
LL2_DWT (640px)	一般画像	52	10152	0.5156	0.6623
	RIC 画像	10204	0	0.9857	0.9865

TABLE 6.3: 電子透かしロバスト性実験結果

640px以上のサイズでリサイズとJPEG圧縮の複合した劣化パターンに対して実験を行う。TABLE 6.4に実験で適用す7個の画像劣化を示す。画像の劣化の具合を例えるため、Figure 6.2に劣化タイプ7番のRICを示す。

TABLE 6.5の実験結果により、リサイズ640ピクセル以上の劣化であっても、読み取りに失敗するケースが存在する。既存RICは画像が1280ピクセルの場合、コードマークサイズが204ピクセルも占めていたが、本研究では24ピクセルしか占めていないほど、

番号	劣化タイプ
1	リサイズ 900px
2	リサイズ 900px × JPEG 50% 圧縮
3	リサイズ 900px × JPEG 30% 圧縮
4	リサイズ 900px × JPEG 30% 圧縮 × リサイズ 1280px × JPEG 30% 圧縮 × リサイズ 640px
5	リサイズ 900px × JPEG 30% 圧縮 × リサイズ 1280px × JPEG 30% 圧縮 × リサイズ 640px × JPEG 30% 圧縮
6	リサイズ 900px × JPEG 30% 圧縮 × リサイズ 1280px × JPEG 30% 圧縮 × リサイズ 640px × JPEG 30% 圧縮 × リサイズ 1000px × JPEG 30% 圧縮
7	リサイズ 900px × JPEG 30% 圧縮 × リサイズ 1280px × JPEG 30% 圧縮 × リサイズ 640px × JPEG 30% 圧縮 × リサイズ 1000px × JPEG 30% 圧縮 × リサイズ 800px × JPEG 30% 圧縮

TABLE 6.4: RIC ロバスト性実験劣化タイプ



Figure 6.2: No. 7の劣化を適用したRIC画像の下部

コードマークが小さくなったことが、ロバスト性を低くしたと言える。読み取り失敗にした画像を分析した結果、全てPSNRが16より未満になる程劣化が激しかった。そのため、本研究は改善したRICをコンテンツ配信に用いるため、PSNRを指標として導入し、画像サイズ640ピクセルかつPSNR16以上の場合に絞ることに

番号	RIC サイズ (px)	平均 PSNR	最小 PSNR	成功率
1	900×17	29.90	28.99	1.0000
2	900×17	22.19	21.27	1.0000
3	900×17	21.08	20.21	1.0000
4	640×12	19.23	18.63	1.0000
5	640×12	17.04	16.43	1.0000
6	1000×19	16.73	16.13	1.0000
7	800×15	16.21	15.68	0.9914

TABLE 6.5: RIC ロバスト性実験結果

した。以上の条件のもと、RIC v0.1 は ISDS に用いることができるロバスト性を持つと証明できた。

6.1.5 比較実験

比較実験は画像にデータを埋め込む既存研究を 2 種類に分け、それぞれの代表となる研究と比較を行う。コードマーク研究からは QR Code [18]、電子透かし研究からは Yesilyurt らの電子透かし [81] と比較を行う。比較は、それぞれの方式でデータを埋め込み、画像を劣化させ、劣化された画像から読み取ったデータが本来のデータと差異なく抽出した場合を成功とみなす。QR Code はレベル 4 の High Error Correction を用い、RIC v0.1 と同じピクセルサイズになるようにした。電子透かしは RIC v1.0 と同じく 900 ピクセルを基準サイズとし、900x900 の領域に 64bit のインデックスデー

たと、64bitのうち最初の部分の48bitのデータを合わせて、冗長化した112bitのデータを埋め込む。実験は以下の手順で行い、その結果をTABLE 6.6に示す。

手順a 64bitサイズの仮想データベースインデックスをランダムに生成する。

手順b 10,000枚から選択した画像に、QR Code、RICそれぞれの手法で埋め込む。

手順c 埋め込まれた画像をリサイズとJPEG圧縮を複合したパターンで劣化させる。

手順d 劣化された画像を読み取り、取り出したデータと本来のデータが一致するか比較する。

TABLE 6.6により、QR Codeと電子透かしはリサイズとJPEG圧縮の影響でデータが破壊されることが証明できる。それに比べ、RICはリサイズ640ピクセル、PSNR 16以上であれば、100%データが読み取れている。3042×4032の画像をLINEに共有すると、1108×1478にリサイズされ、Twitterに共有すると、1536×2048にリサイズされ、Facebookに共有すると、720×960にリサイズされるため、上記の実験結果はRICのみがデジタル画像に適用可能性を持つ唯一の研究であることを示している。

劣化タイプ	QR Code 平均 PSNR	QR Code [18] 成功率	電子透かし [81] 成功率	RIC 成功率
リサイズ 940px	28.68	0.9850	1.0000	1.0000
リサイズ 680px × JPEG 70% 圧縮	20.68	0.9560	0.4243	1.0000
リサイズ 940 × JPEG 30% 圧縮 × リサイズ 1280px × JPEG 30% 圧縮 × リサイ ズ 680px	18.57	0.8720	0.8349	1.0000
リサイズ 940 × JPEG 30% 圧縮 × リサイズ 1280px × JPEG 30% 圧縮 × リサイ ズ 680px × JPEG 30% 圧 縮 × リサイズ 1070px × JPEG 30% 圧縮	16.33	0.6910	0.2146	1.0000

TABLE 6.6: 比較実験結果

第7章

結論

7.1 まとめ

デジタルコンテンツの消費が大変増加しているにもかかわらず、コンテンツの配信インフラには Warner Bros.、Paramount、Disney、Sony のようにコンテンツを制作するコンテンツプロバイダー（プ

ロバイダー)に嬉しくない構造的問題が存在する。既存構造においてプロバイダーは、Netflix、Hulu、Amazonのようなコンテンツを配信するコンテンツプラットフォーマー(プラットフォーマー)にコンテンツを委託し、プラットフォーマーが自分たちのプラットフォーム上でユーザーにコンテンツを配信を行う。また、モバイル環境で配信するコンテンツは、iOSやAndroidのモバイルOSを通して配信する。しかし、モバイルOSとプラットフォーマーを通すコンテンツ配信モデルは、プラットフォーマーへの従属性問題により、トップティアコンテンツが配信されず、プロバイダーはコンテンツの直接販売のルートを探している。

本研究は、プロバイダーのコンテンツ配信にとって、独立した流通インフラを提供するため、超流通の実装モデルであるImage SuperDistribution System (ISDS)を提案し、実装を行なった。ISDSは既存では不可能だったコンテンツの直販ルートを開拓し、市場価値の高いトップティアコンテンツの配信を容易に実施可能にした。また、下記に付加メリットを示す。

- コンテンツプロバイダー側

- プロバイダーはユーザーに直接コンテンツを広告できる。
- プロバイダーはプラットフォーマーのアプリケーションを広告する必要がない。

- プロバイダーはコンテンツ利用規約を自ら指定できる。
- プロバイダーは人気作や最新作を配信できる。
- ユーザー側
 - ユーザーはコンテンツの広告などから 1 クリックでコンテンツを再生できる。
 - コンテンツ再生に会員登録やログインが必要ない。
 - ユーザーはコンテンツは自分のデバイスに画像として保存できる。

本研究では、コンテンツをプラットフォームに依存せず、デバイスから再閲覧を可能にする Robust Index Code を提案し、実験により RIC の有用性を確かめ、RIC を用いて ISDS を実装した。

7.2 研究の貢献

本研究の貢献は超流通を基盤とした新たなコンテンツ直販モデルの提案及び実装である。ISDS の実装における技術的困難さの一つは、プラットフォームに依存しないメディアであるデジタル画像をコンテンツと紐づけることである。また、ISDS のサービス実現における困難さは、プロバイダーは自分たちのプラットフォームを作りたがるが、そのシステムは独立型ではなく、既存

プラットフォームと同様の一つのプラットフォームであるということやシステムを開発及び運用にあたり莫大なコストがかかってしまう。従って、本研究はRICにより技術的困難さを解決し、独立型インフラであるISDSの実装によりサービス実現の困難さを解決した。その結果、ISDSはプロバイダーがコンテンツを直接販売できる新たな流通経路として使用すると期待できる。

7.3 今後の課題

ロバスト性の拡張

6.1.4節の実験により、RICは640ピクセル以上及びPSNR16以上であれば100%読み取れるが、その以外の場合、読み取れないケースが生じる。デザインを改善したことにより、画像のサイズが小さくなると、データが埋め込まれている領域のサイズも極めて小さくなり、JPEG圧縮で円の中心に色の変化が生じやすくなる。従って、色の変化が生じないように防ぐ方法あるいは色の変化を復元する手法の提案が必要である。

ユーザーの一意性

ISDSはデバイスの固有IDを用いて、ユーザーを一意に識別している。コンテンツを購入したユーザーはそのデバイスでのみ再生することができる。従って、他のデバイスで購入した

コンテンツは再生できない問題がある。この点はFacebookログインなどのSNSログインによって解決できるが、ユーザーがログインを行わなければいけないことにより、利便性が下がる。従って、一人のユーザーが複数のデバイスで視聴できる一意性を担保できるアルゴリズムが必要である。

画像の不正修正対策

本研究はポスター画像にRICを埋め込みコンテンツ配信を提案している。しかし、悪意のあるユーザーがRIC画像のポスターを不正修正し、インターネット上に再配信する可能性がある。既存RICでは画像の特徴をコンテンツと一緒にリモートデータベースに保存しているため、不正修正した画像からコンテンツへのアクセスはある程度防げるが、限界がある。そのため、正確に不正修正を発見できるアルゴリズムの考察が必要である。

謝辞

本研究を進めるにあたり、ご協力頂いたPulit株式会社および権藤研究室の皆様へ深く感謝致します。

新しい分野に挑戦することは非常に難しく、悩む毎日でしたが、無事本稿を出す事が出来たのは適切な助言をくださった権藤教授のおかげだと思います。

協力して頂いた皆様へ心からの感謝の気持ちと御礼を申し上げたく、謝辞にかえさせていただきます。

本論文の図で用いられたアイコンは www.flaticon.com の Yannick, Smashicons, Freepik, smalllikeart, photo3idea-studio により作られたものと、icons8のものとなります。(Icons made by Yannick, Smashicons, Freepik, smalllikeart, photo3idea-studio from www.flaticon.com and App Icons designed by icons8 <https://icons8.com/>)

参考文献

- [1] 総務省, “平成 29 年版情報通信白書 — 通信系コンテンツ市場の動向,” [Online]. Available: <http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h29/html/nc261830.html>, 2017.
- [2] M. Kim, K. Lee, and K. Gondow, “Distribute digital contents within digital images on the mobile,” in *Proc. of the 9th Int. Conf. on E-Education, E-Business, E-Management and E-Learning*, ser. IC4E '18, New York, NY, USA, 2018, pp. 35–39.
- [3] Appleinsider, “Apple’s App Store generated over \$10 billion in revenue for developers in record 2014,” [Online]. Available: <http://appleinsider.com/articles/15/01/08/apples-app-store-generated-over-10-billion-in-revenue-for-developers-in-record-2014>, 2014.
- [4] T. Spangler, “YouTube Standardizes Ad-Revenue Split for All Partners, But Offers Upside Potential,” [Online]. Available: <http://variety.com/2013/digital/news/youtube-standardizes->

ad-revenue-split-for-all-partners-but-offers-upside-potential-1200786223/, 2013.

- [5] InvestorPlace, “No Disney (DIS) Movies, No Problem for Netflix (NFLX),” [Online]. Available: <https://investorplace.com/2017/09/no-disney-dis-no-problem-netflix-nflx/#.WdhRxpvAN25>, 2017.
- [6] R. Mori and M. Kawahara, “Superdistribution: The Concept and the Architecture,” *IEICE Trans.*, vol. 73, no. 7, pp. 1133–1146, Jul. 1990.
- [7] P. Korus, J. Białas, and A. Dziech, “Towards practical self-embedding for JPEG-compressed digital images,” *IEEE Trans. Multimedia*, vol. 17, no. 2, pp. 157–170, 2015.
- [8] T. Thomas, S. Emmanuel, A. V. Subramanyam, and M. S. Kankanhalli, “Joint Watermarking Scheme for Multiparty Multilevel DRM Architecture,” *IEEE Trans. Information Forensics and Security*, vol. 4, no. 4, pp. 758–767.
- [9] A. Motahari and M. Adjouadi, “Barcode Modulation Method for Data Transmission in Mobile Devices,” *IEEE Trans. Multimedia*, vol. 17, no. 1, pp. 118–127, 2015.

- [10] A. Briassouli and P. Tsakalides, “Hidden messages in heavy-tails: DCT-domain watermark detection using alpha-stable models,” *IEEE Trans. Multimedia*, vol. 7, no. 4, pp. 700–715, 2005.
- [11] C.-T. Yen, H.-T. Liaw, N.-W. Lo, T.-C. Liu, and J. Stu, “Transparent Digital Rights Management System with Superdistribution.” *BWCCA*, 2010.
- [12] M. Kim, K. Lee, K. Gondow and J.-I. Imura, “Robust Index Code with Digital Images on the Internet,” in *Proc. 13th Int. Joint Conf. e-Business and Telecommunications*, vol. 5: SIGMAP, 2016, pp. 28–37.
- [13] K. Lee, “Robust Index Code,” JP 2016-177743 A, Oct 2016.
- [14] M. Kim, K. Lee, K. Gondow, and J.-i. Imura, “Robust Index Code to Distribute Digital Images and Digital Contents Together,” *IEICE Trans. on Information and Systems*, vol. E101.D, no. 9, pp. 2179–2189, 2018.
- [15] Apple, “App Store Review Guidelines,” [Online]. Available: <https://developer.apple.com/app-store/review/guidelines/#in-app-purchase>, 2016.

- [16] Google Play, “Monetization and Ads,” [Online]. Available: <https://play.google.com/intl/en/about/monetization-ads/>, 2018.
- [17] Wikipedia, “電子透かし,” [Online]. Available: <https://ja.wikipedia.org/wiki/電子透かし>, 2004.
- [18] D. W. Incorporated., “About QR Code,” [Online]. Available: <http://www.qrcode.com/>, 2008.
- [19] K. A. H. Nurwono and R. Kosala, “Color quick response code for mobile content distribution.” *MoMM*, pp. 267–271, 2009.
- [20] M. Research, “High Capacity Color Barcodes,” [Online]. Available: <http://research.microsoft.com/en-us/projects/hccb/>, 2010.
- [21] M. Jiansheng, L. Sukang, and T. Xiaomei, “A digital watermarking algorithm based on DCT and DWT,” *Int. Symposium on Web Information Systems and Applications, ser. WISA ’ 09*, 2009.
- [22] M. Barni, F. Bartolini, V. Cappellini, and A. Piva, “A DCT-domain system for robust image watermarking,” *Signal Processing*, vol. 66, no. 3, pp. 357–372, May 1998.
- [23] Wikipedia, “Superdistribution,” [Online]. Available: <https://en.wikipedia.org/wiki/Superdistribution>, 2013.

- [24] R. Upathilake, Y. Li, and A. Matrawy, “A classification of web browser fingerprinting techniques,” in *2015 7th Int. Conf. on New Technologies, Mobility and Security (NTMS)*. IEEE, 2015, pp. 1–5.
- [25] Amazon Kindle, “eBook Royalty Options,” [Online]. Available: https://kdp.amazon.com/en_US/help/topic/G200644210, 2018.
- [26] Broadband TV News, “US SVOD subscriptions to reach 270 million,” [Online]. Available: <https://www.broadbandtvnews.com/2019/04/17/us-svod-subscriptions-to-reach-270-million/>, 2019.
- [27] V. K. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, and Z.-L. Zhang, “Unreeling Netflix: Understanding and Improving Multi-CDN Movie Delivery ,” *INFOCOM*, pp. 1620–1628, 2012.
- [28] BGR, “Sorry, ‘Friends’ fans: WarnerMedia has decided to pull its shows from Netflix,” [Online]. Available: <https://bgr.com/2019/05/14/netflix-shows-leaving-friends-office-warnermedia/>, 2019.
- [29] I. Premium Platform Japan, “Paravi,” [Online]. Available: <https://www.paravi.jp>, 2018.

- [30] I. Fuji Television Network, “FOD,” [Online]. Available: <http://fod.fujitv.co.jp/s/>, 2005.
- [31] T. A. Co.Ltd, “TOEI ANIMATION OnDemand,” [Online]. Available: <https://taod.jp>, 2013.
- [32] MarkeZine, “動画配信市場の規模推計、前年比19.5%増Netflixなどが牽引【GEM Partners調査】,” [Online]. Available: <https://markezine.jp/article/detail/30433>, 2019.
- [33] internet.org, “A Focus on Efficiency,” [Online]. Available: <http://internet.org/efficiencypaper>, 2013.
- [34] W3Techs, “Usage of image file formats for websites,” [Online]. Available: http://w3techs.com/technologies/overview/image_format/all, 2015.
- [35] PhotoShelter, “Why Facebook Photos Look so Bad, and the DIY Solution to Fix It,” [Online]. Available: <http://blog.photoshelter.com/2014/01/facebook-photos-look-bad-diy-solution-fix/>, 2014.
- [36] Wikipedia, “Barcode,” [Online]. Available: <https://en.wikipedia.org/wiki/barcode>, 2015.
- [37] Wikipedia, “二次元コード,” [Online]. Available: <https://ja>.

- wikipedia.org/wiki/二次元コード, 2018.
- [38] Wikipedia, “カメレオン・コード,” [Online]. Available: <https://ja.wikipedia.org/wiki/カメレオン・コード>, 2018.
- [39] B.-N. Li, M.-C. Dong, and M. I. Vai, “From Codabar to ISBT 128: Implementing Barcode Technology in Blood Bank Automation System,” in *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual Int. Conf. of the. IEEE, 2005*, pp. 542–545.
- [40] D. Savir and G. J. Laurer, “The characteristics and decodability of the Universal Product Code symbol,” *IBM Systems Journal*, vol. 14, no. 1, pp. 16–34, 1975.
- [41] C.-S. Woo, J. Du, and B. Pham, “Performance Factors Analysis of a Wavelet-based Watermarking Method.” *ACSW Frontiers*, pp. 89–97, 2005.
- [42] M. Jiansheng, L. Sukang, and T. Xiaomei, “A Digital Watermarking Algorithm Based On DCT and DWT,” in *Proc. Int. Symp. Web Information Systems and Applications*, ser. WISA '09. Academy Publisher, May 2009, pp. 104–107.
- [43] C. Wang, J. Ni, J. Huang, R. Zhang, and M. Huang, “Robust and

- High Capacity Image Watermarking Based on Jointly Coding and Embedding Optimization,” in *Information Hiding*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007, vol. 4567, pp. 65–79.
- [44] G. K. Wallace, “The JPEG still picture compression standard,” *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.
- [45] J. Jeswani and T. Sarode, “An Improved Blind Color Image Watermarking using DCT in RGB Color Space,” *Int. Journal of Computer Applications*, vol. 92, no. 14, pp. 50–56, 2014.
- [46] R. Eswaraiah, E. Sreenivasa Reddy, “Robust Watermarking Method for Color Images using DCT Coefficients of Watermark,” *Global Journal of Computer Science and Technology*, Aug. 2012.
- [47] J. Jiang and A. Armstrong, “Data hiding approach for efficient image indexing,” *Electronics Letters*, vol. 38, no. 23, pp. 1424–1425, Nov 2002.
- [48] F. Huang, X. Qu, H.-J. Kim, and J. Huang, “Reversible Data Hiding in JPEG Images,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 26, no. 9, pp. 1610–1621, 2016.

- [49] G. Jayamurugan, “Lossless and Reversible Data Hiding in Encrypted Images with Public Key Cryptography,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 26, no. 9, pp. 1622–1631, 2016.
- [50] W. Zhang, H. Wang, D. Hou, and N. Yu, “Reversible Data Hiding in Encrypted Images by Reversible Image Transformation.” *IEEE Trans. Multimedia*, 2016.
- [51] Z. Qian, X. Zhang, and S. Wang, “Reversible data hiding in encrypted JPEG bitstream,” *IEEE Trans. Multimedia*, vol. 16, no. 5, pp. 1486–1491, 2014.
- [52] W.-L. Tai, C.-M. Yeh, and C.-C. Chang, “Reversible Data Hiding Based on Histogram Modification of Pixel Differences.” *IEEE Trans. Circuits and Systems for Video Technology*, 2009.
- [53] W. Hong, T.-S. Chen, and C.-W. Shiu, “Reversible data hiding for high quality images using modification of prediction errors.” *Journal of Systems and Software*, vol. 82, no. 11, pp. 1833–1842, 2009.
- [54] Q. Liu, R. Safavi-Naini, and N. P. Sheppard, “Digital Rights Management for Content Distribution.” *ACSW*, 2003.

- [55] Marco M. Hurtado, James C. Mahlbacher and Richard L. Spagna, “Super-distribution of protected digital content,” US09711431, Jan 2006.
- [56] Dong-seok Kang, “Digital contents superdistribution system and method of distributing digital contents,” US09783560, Oct 2006.
- [57] Thomas Dowdy, Jeffrey L. Robbin, Guy L. Tribble and David Heller, “Method and system for secure network-based distribution of content,” US7895661B2, Feb 2011.
- [58] Itseez, “OpenCV,” [Online]. Available: <http://opencv.org>, 2000.
- [59] Google Android, “Android NDK,” [Online]. Available: <https://developer.android.com/ndk>, 2009.
- [60] B. Furht, “A Survey of Multimedia Compression Techniques and Standards. Part I: JPEG Standard,” *Real-Time Imaging*, vol. 1, no. 1, pp. 49–67, 1995.
- [61] Wikipedia, “チェックサム,” [Online]. Available: <https://ja.wikipedia.org/wiki/チェックサム>, 2015.
- [62] S. Xiang, H.-J. Kim, and J. Huang, “Histogram-based image hashing scheme robust against geometric deformations,” in *Pro-*

- ceedings of the 9th Workshop on Multimedia & Security*, ser. MM&Sec '07, 2007, pp. 121–128.
- [63] C. Mingzhi, L. Yan, Z. Yajian, and L. Min, “A Combined DWT and DCT Watermarking Scheme Optimized Using Genetic Algorithm,” *Journal of Multimedia*, vol. 8, no. 3, pp. 1–7, Jun. 2013.
- [64] A. Afuah, *Business Models: A Strategic Management Approach*, Jan. 2003.
- [65] 谷井 良, “ビジネスモデルイノベーションによる事業創造のデザイン,” *明星大学経営学研究紀要*, vol. 9, pp. 89–99, Mar. 2016.
- [66] マーク W ジョンソン, クレイトン M クリステンセン and ヘニング カガーマン, “ビジネスモデル・イノベーションの原則,” *DAIAMOND ハーバード・ビジネス・レビュー*, Apr. 2009.
- [67] Google, “Google AdSense,” [Online]. Available: <https://www.google.com/adsense/>, 2019.
- [68] 平川 学, “著作権保護における電子透かしサービスモデルの有効性検証,” in *経営情報学会 2008年春季全国研究発表大会*, Jun. 2008, pp. 418–421.
- [69] K.-S. Chen and M.-H. Chen, “EC 2.0: Can you get profit by

- writing blog? An empirical study in Google AdSense,” in *Int. Conf. on Technology Management for Global Economic Growth*, ser. PICMET '10, Jul. 2010, pp. 1–7.
- [70] A. Tagaris, A. Giakoumaki, L. Karle, and D. Koutsouris, “Watermarking SDK Implementation to Facilitate Integration in a Secure Healthcare Environment,” in *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2006, pp. 3262–3265.
- [71] Inc.com, “The 10 Most Downloaded Apps in the World All Have This 1 Amazing Thing in Common,” [Online]. Available: <https://www.inc.com/gene-marks/the-worlds-10-most-downloaded-apps-all-have-this-one-amazing-thing-in-common.html>, 2017.
- [72] J. Tunnecliff, J. Weiner, J. E. Gaida, J. L. Keating, P. Morgan, D. Ilic, L. Clearihan, D. Davies, S. Sadasivan, P. Mohanty, S. Ganesh, J. Reynolds, and S. Maloney, “Translating evidence to practice in the health professions: a randomized trial of Twitter vs Facebook,” *Journal of the American Medical Informatics Association*, p. ocw085, 2016.

- [73] J-CAST テレビウォッチ, “2018年「ドラマ視聴率ランキング」女王・米倉涼子抜いた松本潤！ ベストテンすべてテレ朝系とTBS系,” [Online]. Available: <https://www.j-cast.com/tv/2018/12/26346712.html>, 2018.
- [74] 石川 俊之, 浅田 孝利, 栗原 潤, 中村 一郎, 塩澤 秀和, and 小泉 寿男, “自動選考機能を持つ部品調達ビジネスモデルと構築評価,” マルチメディア通信と分散処理ワークショップ論文集, vol. 2001, no. 13, pp. 133–138, Oct. 2001.
- [75] Google, “Angular,” [Online]. Available: <https://angular.io>, 2016.
- [76] Pivotal Software, “Spring Framework,” [Online]. Available: <https://spring.io/projects/spring-framework>, 2015.
- [77] The Apache Software Foundation, “Apache Tomcat 8,” [Online]. Available: <https://tomcat.apache.org/tomcat-8.5-doc/index.html>, 2015.
- [78] Oracle Corporation, “MySQL,” [Online]. Available: <https://www.mysql.com/jp/>, 2015.
- [79] K. Connors, M. Connors, and J. Seemann, “morgueFile,” [Online]. Available: <http://www.morguefile.com>, 1996.

- [80] A. Hore and D. Ziou, “Image Quality Metrics: PSNR vs. SSIM,” in *Pattern Recognition (ICPR), 2010 20th Int. Conf. on.* IEEE, 2010, pp. 2366–2369.
- [81] M. YESILYURT, Y. YALMAN, and A. T. OZCERIT, “A New DCT Based Watermarking Method Using Luminance Component,” *Electronics and Electrical Engineering*, vol. 19, no. 4, 2013.

付録 A

ユーザーの動画再生フロー

ユーザーがプラットフォームアプリを持っていない状況で、コンテンツの広告を見て、動画を再生してみたいユースケースの場合を説明する。本章ではiPhoneの場合を想定する。

ステップ 1

Figure A.1のように、アプリストアを開く。

ステップ 2

Figure A.2のように、検索タブを選択する。

ステップ 3

Figure A.3のように、プラットフォームのアプリを検索する。

ステップ 4

Figure A.4のように、アプリのインストールを待つ。

ステップ 5

Figure A.5のように、アプリを開き、会員登録及び決済手段を登録する。

ステップ 6

最後に、Figure A.6のように、本来見たかった動画を再生する。

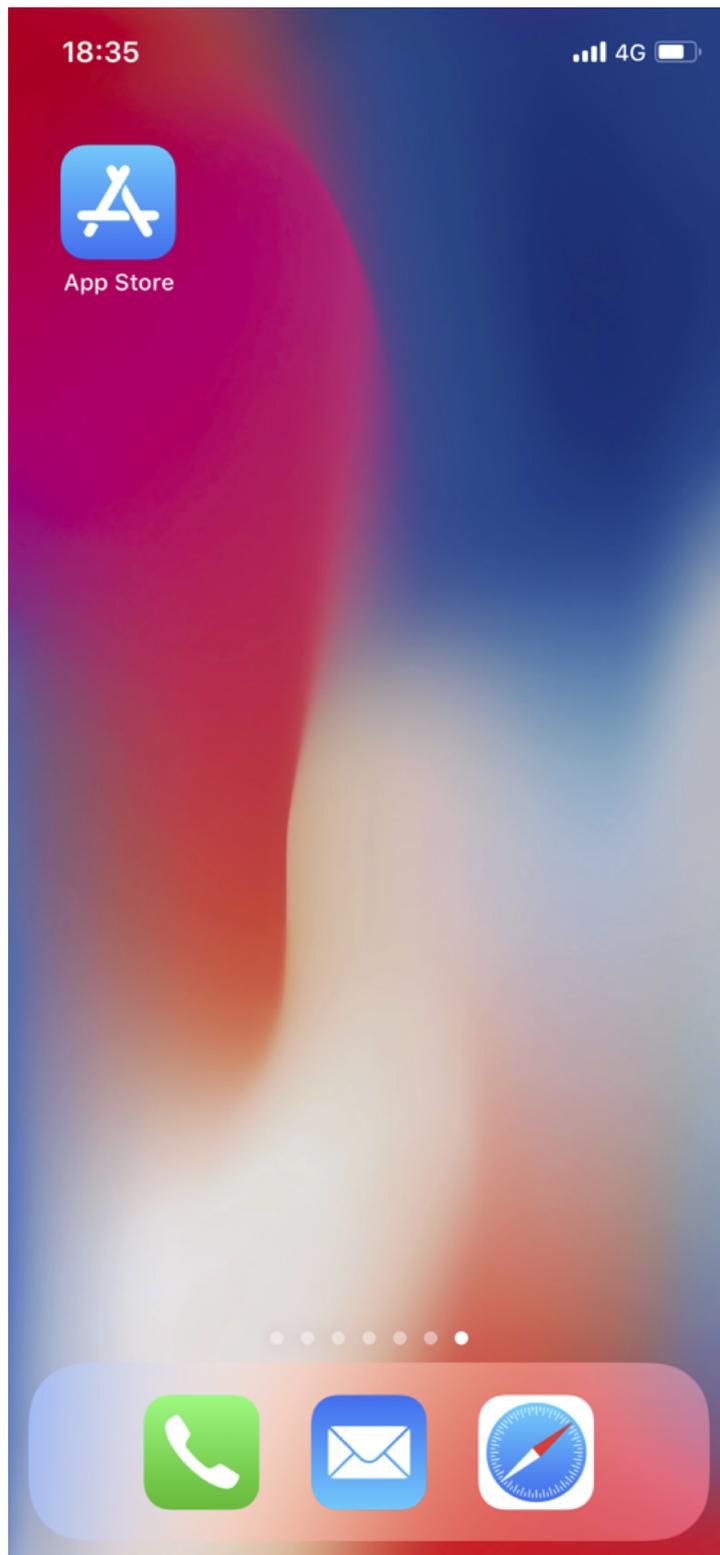


Figure A.1: プラットフォーマーで動画再生手順ステップ1

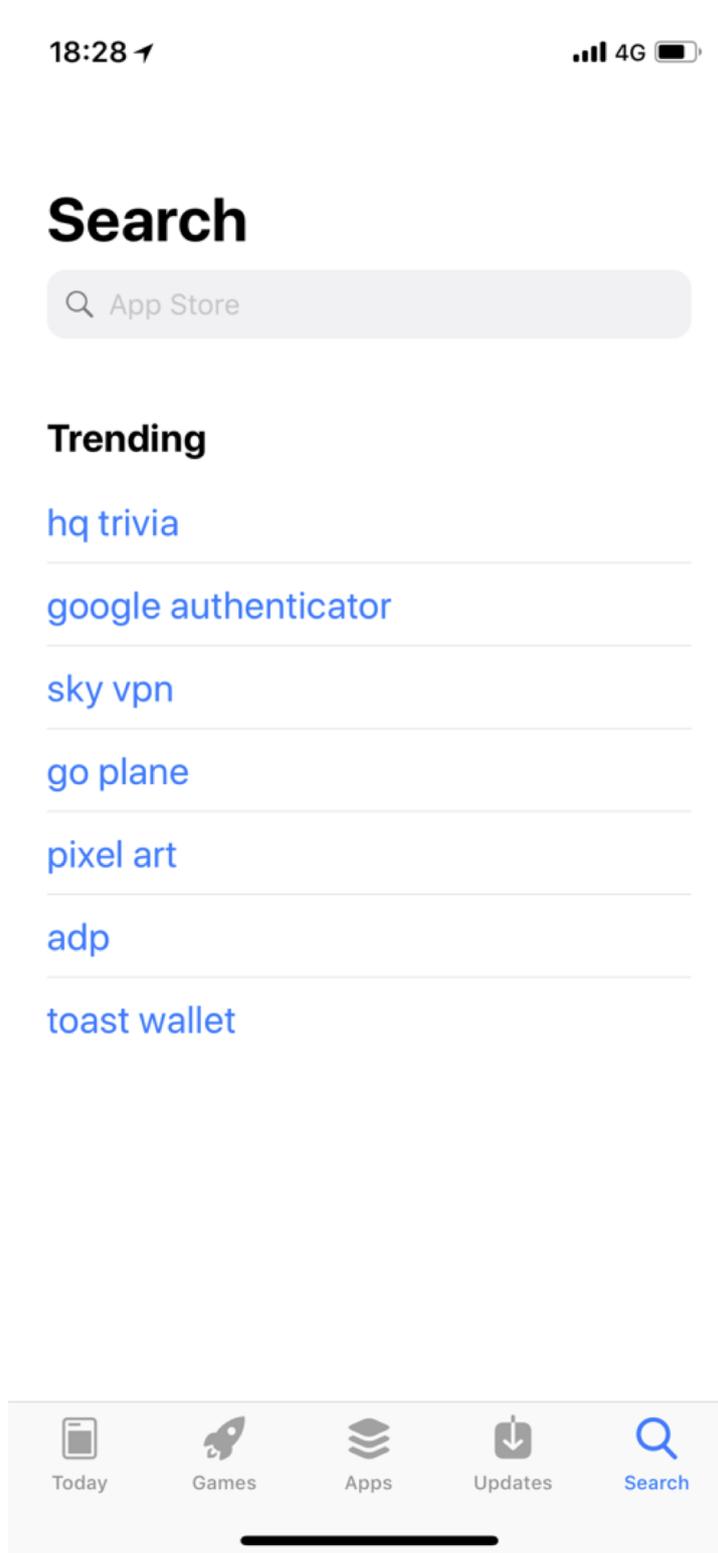


Figure A.2: プラットフォーマーで動画再生手順ステップ2

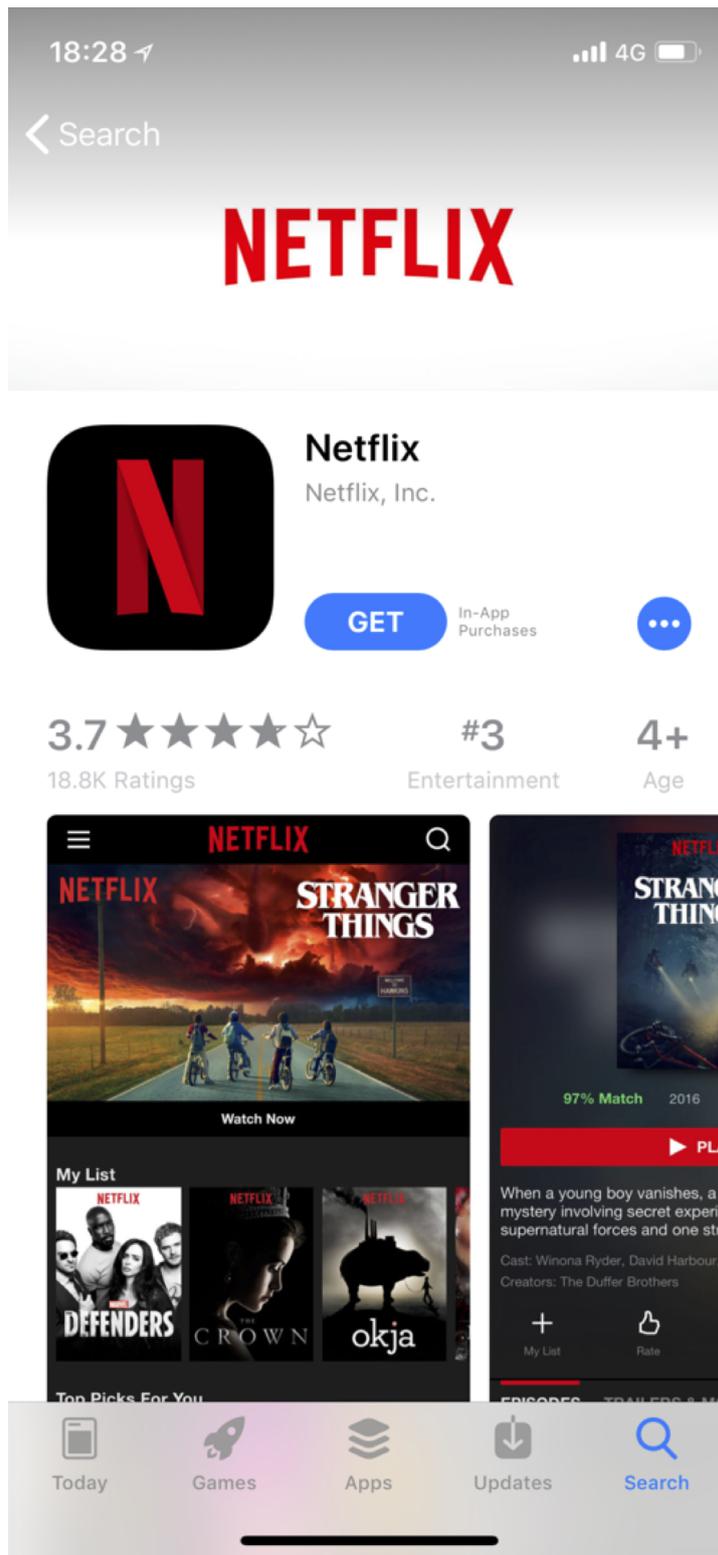


Figure A.3: プラットフォーマーで動画再生手順ステップ3

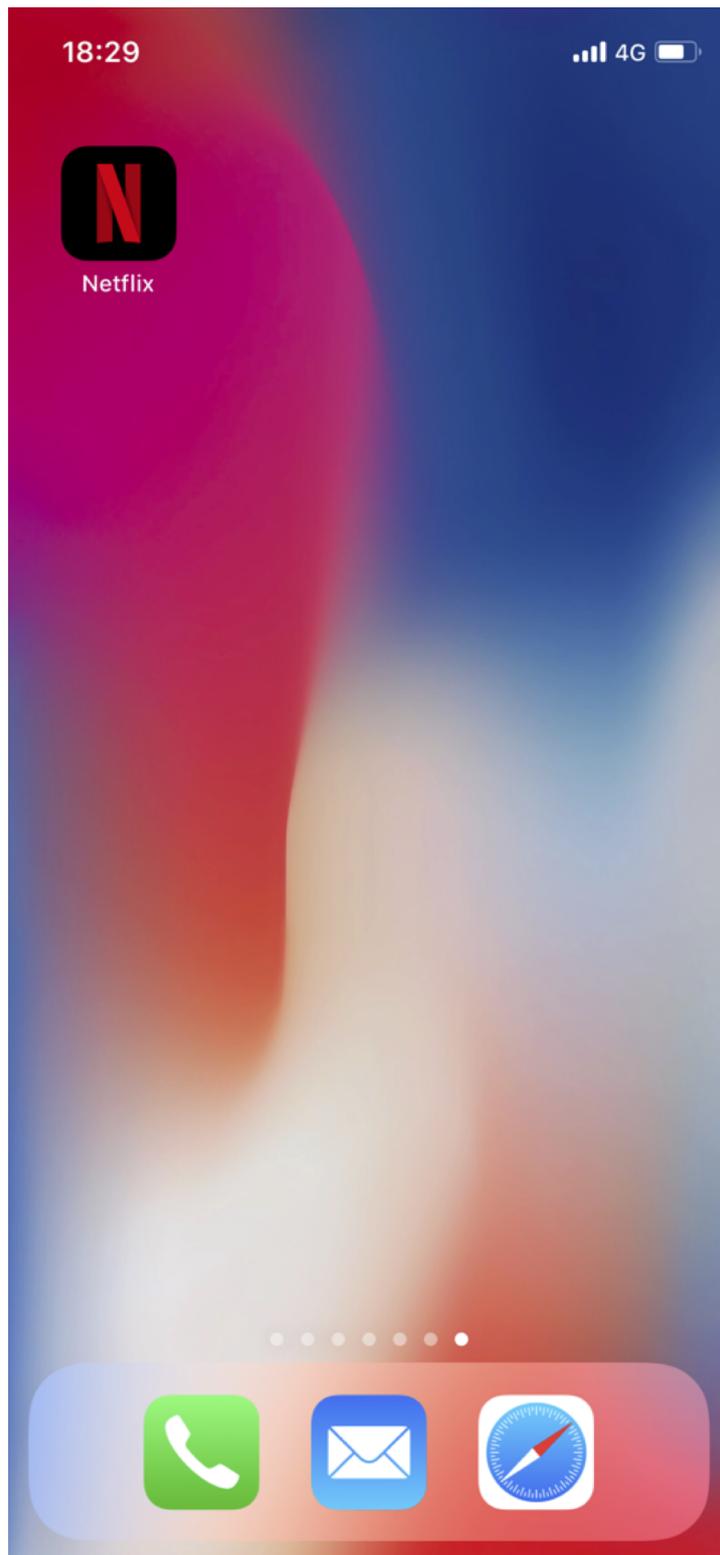


Figure A.4: プラットフォーマーで動画再生手順ステップ4

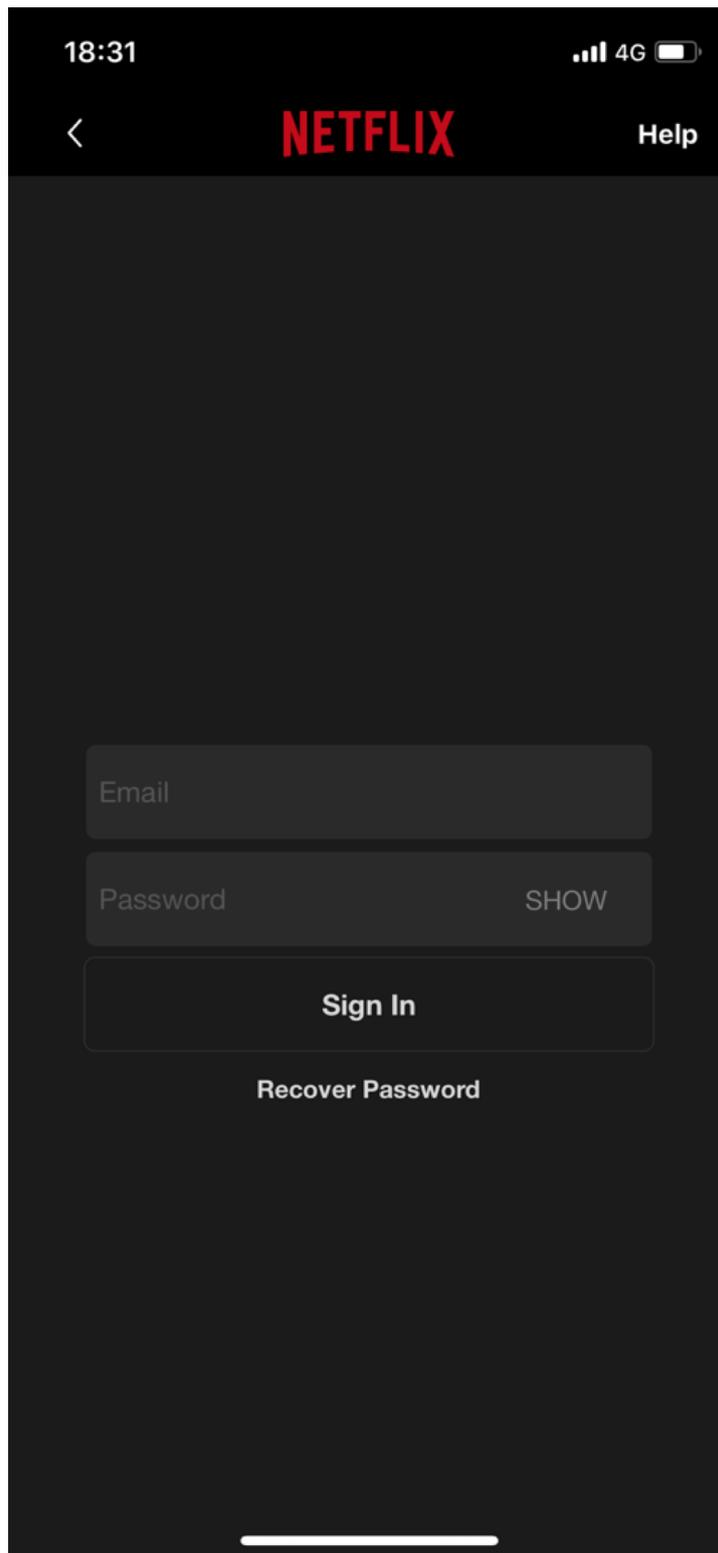


Figure A.5: プラットフォーマーで動画再生手順ステップ5



Figure A.6: プラットフォーマーで動画再生手順ステップ6

付録 B

電子透かし実装コード

本章では、RIC デザイン改善で用いた電子透かし手法におき、DWTに関する実装コードを示す。

Listing B.1: 離散ウェーブレット変換 (DWT) コード (DWTManager.cpp)

```
#include "DWTManager.h"
#include "HaarTransform.h"

using namespace std;
using namespace cv;

DWTManager::DWTManager(const cv::Mat& mat)
{
    this->mat = mat;
    cv::split(mat, bgr_planes);
}

DWTManager::~DWTManager()
{
    if(!dwt1Mat.empty()) {
        dwt1Mat.release();
    }
    for(pair<Mat, Mat>& pair : dwtMats) {
        if(!pair.first.empty()) {
            pair.first.release();
        }
    }
}
```

```

        if(!pair.second.empty()) {
            pair.second.release();
        }
    }
}

cv::Mat DWTManager::applyDWT()
{
    if(!dwtMats.empty()) {
        dwtMats.clear();
    }

    const int marginX = Constants.WATERMARK_LEFT_MARGIN;
    const int marginY = Constants.WATERMARK_BOTTOM_MARGIN
        ;
    const int blockLength = Constants.DCT_BLOCK_LENGTH;

    const cv::Point2i u(Constants.DWT_DCT_COEF_ONE_ROW,
        Constants.DWT_DCT_COEF_ONE_COL);
    const cv::Point2i v(Constants.DWT_DCT_COEF_TWO_ROW,
        Constants.DWT_DCT_COEF_TWO_COL);

    int multiplier = Constants.WATERMARK_DWT_LEVEL == 0 ?
        1 : pow(2, Constants.WATERMARK_DWT_LEVEL+1);

    const int dwtMarginX = (marginX*2+blockLength);
    const int dwtMarginY = (marginY*2+blockLength);
    const int dwtMarginRight = (marginX*2+blockLength);
    const int dwtMarginBottom = (marginY*2+blockLength);

    blueMat = bgr_planes[0];

    // DWT //////////////////////////////////

    blueMat.convertTo(blueFloatMat, CV_64FC1);
    int width = blueFloatMat.cols - dwtMarginX -
        dwtMarginRight + 1;
    while(--width % multiplier != 0);

```

```

int height = blueFloatMat.rows - dwtMarginY -
    dwtMarginBottom + 1;
while(--height % multiplier != 0);
dwtPlaneMat = Mat(blueFloatMat, Rect(dwtMarginX,
    blueFloatMat.rows-dwtMarginBottom-height, width,
    height));

dwt1Mat = HaarTransform::cvHaarWavelet(dwtPlaneMat,
    1);
Mat cLL1Mat(dwt1Mat, Rect(0,0,dwt1Mat.cols/2,dwt1Mat.
    rows/2));

dwtMats.reserve(Constants.WATERMARK_DWT_LEVEL);
dwtMats.push_back(make_pair(dwt1Mat, cLL1Mat));

for(int i=0;i<Constants.WATERMARK_DWT_LEVEL-1;++i) {
    // DWT with Haar transform
    Mat dwtMat = HaarTransform::cvHaarWavelet(dwtMats
        [i].second, 1);
    Mat cLLMat(dwtMat, Rect(0,0,dwtMat.cols/2,dwtMat.
        rows/2));
    dwtMats.push_back(make_pair(dwtMat, cLLMat));
}

return dwtMats.back().second;
}

cv::Mat DWTManager::applyInverseDWT()
{
    if(dwtMats.empty()) {
        return Mat();
    }

    for(vector<pair<Mat, Mat>>::reverse_iterator itr=
        dwtMats.rbegin();itr!=dwtMats.rend()-1;++itr) {
        Mat embeddedDwtMat = HaarTransform::
            cvInvHaarWavelet(itr->first, 1, HaarTransform
                ::NONE);
    }
}

```

```

        addWeighted((itr+1)->second, 0, embeddedDwtMat,
            1, 0, (itr+1)->second);
    }
    Mat dwtEmbeddedMat = HaarTransform::cvInvHaarWavelet(
        dwt1Mat, 1, HaarTransform::NONE);
    addWeighted(dwtPlaneMat, 0, dwtEmbeddedMat, 1, 0,
        dwtPlaneMat);
    blueFloatMat.convertTo(blueMat, CV_8UC1);

    cv::Mat embeddedMat;
    merge(bgr_planes, embeddedMat);

    return embeddedMat;
}

```

Listing B.2: "GAをもとにしたパラメータ探しアルゴリズムコード"

```

int WatermarkManager::embedWatermarkWithHighStability(cv
    ::Mat& mat)
{
    int dwtC = FSolutionConstants::
        DWT_DCT_COEF_HIGH_GAIN_FACTOR;
    if(applyHighStability[0]) {
        for(;dwtC<=FSolutionConstants::
            DWT_DCT_COEF_LIMIT_GAIN_FACTOR;dwtC+=10) {
            // embed watermark and check the image has
            robustness on JPEG Comp.
            bool dwtEasyDamaged =
                checkImageIsEasyToDamage(mat, dwtC);
            if(!dwtEasyDamaged) {
                return dwtC;
            }
        }
    }
    return dwtC;
}

```

付録 C

ISDS コンソール利用シーン

本章では、コンテンツプロバイダーがコンテンツをアップロードし、コンテンツの利用規約が設定でき、URLとRIC画像をもらえるコンソールプログラムの利用シーンを示す。

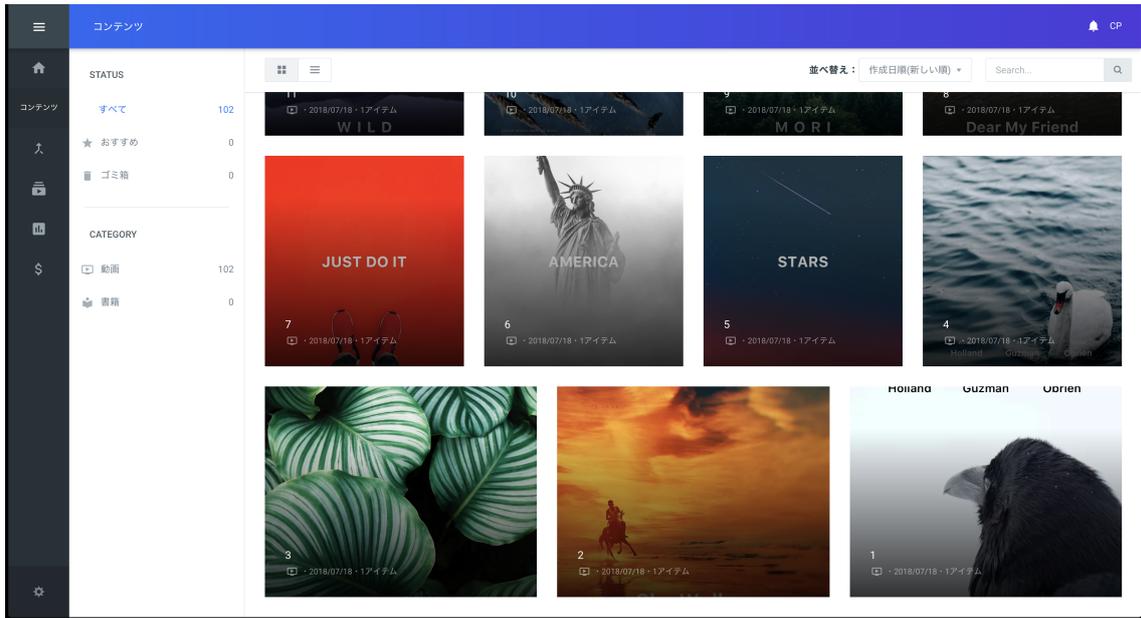


Figure C.1: ISDS コンソール : コンテツツリスト

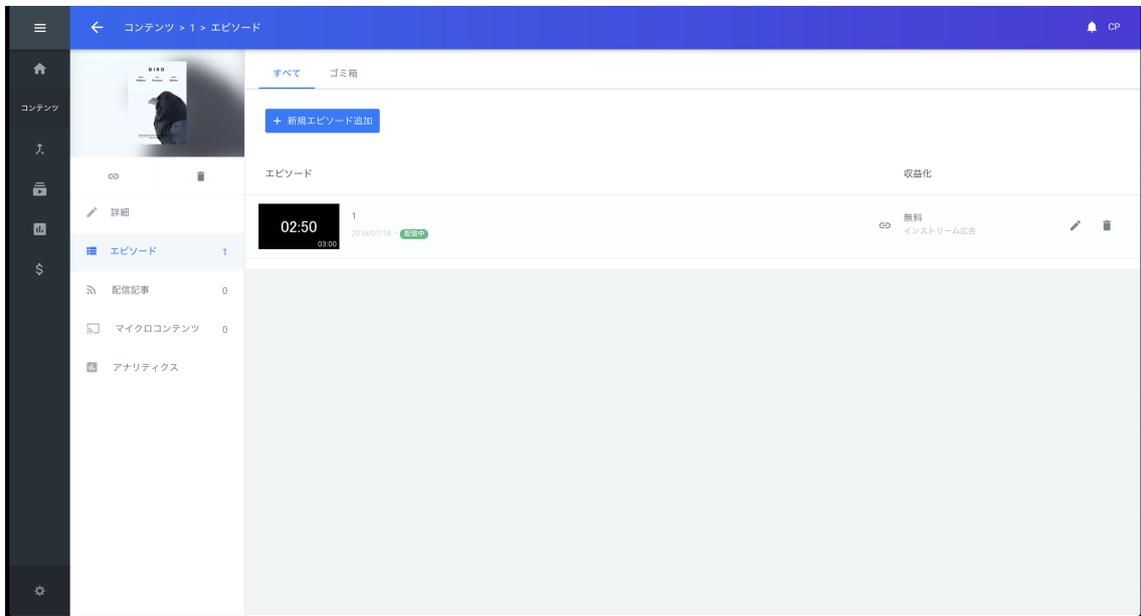


Figure C.2: ISDS コンソール : コンテツツ内のエピソードリスト

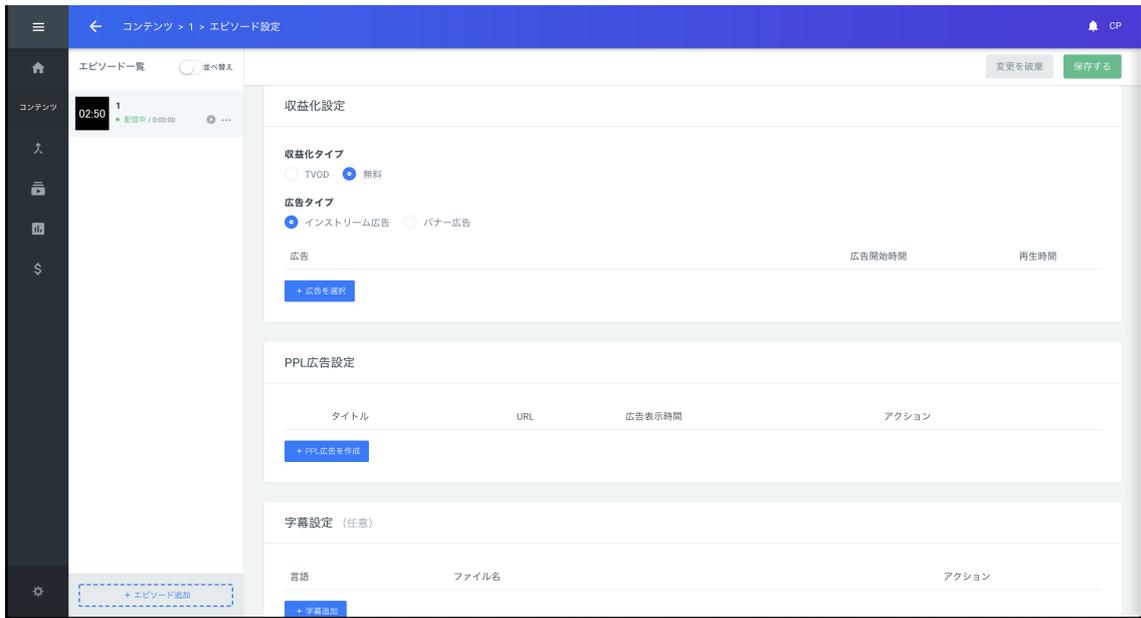


Figure C.3: ISDS コンソール : コンテンツ利用規約設定

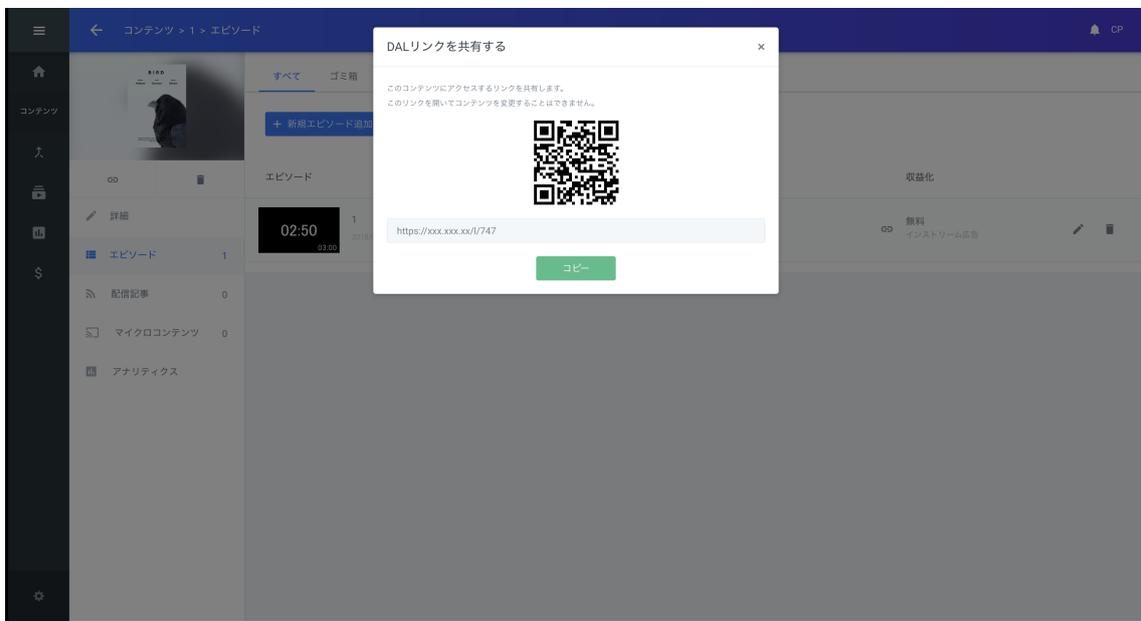


Figure C.4: ISDS コンソール : コンテンツと紐づく URL 取得