

動画像解析による漫画の自動生成に関する研究

福元清一^{†1} 加納実樹^{†2} 中野寿子^{†2} 湊田孝康^{†1}

概要: スマートデバイスやインターネットの世界的な普及に伴い、動画投稿サイトの利用者が急激に増えており、近年では誰でも気軽に動画像を撮影・投稿・閲覧することが可能になっている。また、サブカルチャーの発展もあり、現代では日本はもちろん海外でも漫画が日常に浸透してきている。そこで本研究では、現代のニーズにマッチングした動画像処理の研究として、動画像から動画像解析によって漫画の自動生成を行う手法を考えた。また、動画から漫画の形に変換することは情報の圧縮方法になると考えられる。今回は動画像から漫画のコマに適したシーンを切り出すことを中心に研究し、主にオプティカルフローといった動きの特徴量などの情報から漫画に適したコマの候補を切り出し、コマを選択・配置して漫画を自動生成する手法を検討した。

キーワード: コンピュータビジョン, 画像特徴抽出, 動画像解析

Study on automatic generation of the comics using movie analysis

KIYOKAZU FUKUMOTO^{†1} MIKI KANOU^{†2}
HISAKO NAKANO^{†2} TAKAYASU FUCHIDA^{†1}

Abstract: With the spread of smart devices and the Internet worldwide, users of video posting sites are rapidly increasing, and in recent years anyone can easily take, post, and view moving images. Also, with the development of subculture, cartoons are penetrating in daily life in modern Japan as well as overseas. Therefore, in this research, as a study of moving image matching the needs of modern times, a method of automatically generating cartoons from moving images through moving image analysis was considered. Also, it is considered that converting from moving images to cartoon forms becomes a method of compressing information. In this time, we focus on cutting out scenes suitable for manga scenes from moving images, cut candidates suitable for manga from information such as the movement feature amount such as optical flow, mainly, select and place frames. We developed a method to automatically generate cartoons.

Keywords: Computer vision, Image feature extraction, Movie analysis

1. 研究背景と目的

近年ではインターネットの普及に伴い、YouTube やニコニコ動画等の動画投稿サイトの利用者が急激に増えている。それに加えてスマートフォンといったスマートデバイスの世界的な普及もあり、多くの人がどこでも気軽に動画像を撮影・投稿・閲覧することが可能になっている。また、ホームビデオや監視カメラの映像、アニメや実写映画といった作品等も含めて考えると世の中に膨大な量の動画像がある。これらのことから動画像処理は現代において関心を集め、盛んに研究が行われている分野であると言える。

また、現代において「漫画」は見かけないことはないほどに浸透している人気のコンテンツである。日本における近年のサブカルチャーの発展を考慮すると、漫画は現代において注目を浴びている日本文化の一つと考えられる。

既に静止画像を漫画風に変換する技術は数多くあり、アプリとして気軽に利用できるものもある。漫画風の画像

処理技術は、画像処理の知識のない人々にも利用できるコンテンツになっている。

以上のことから、動画像解析による漫画の自動生成に関する研究は現代のニーズにマッチングした動画像処理の研究であると考え、これらの背景から本研究を行った。

本研究の目的のひとつは動画像のもつ情報を圧縮する方法としての漫画自動生成の手法を提案することである。動画を再生して視聴するには、5分の動画であれば5分間の時間を要する。多少の早送りをしても、どのような人でも一定以上のまとまった時間が必要になる。漫画という形であれば、読む人の自由なペースで読むことができる。要する時間は動画を視聴するよりも短く済むと思われる。また、動画はデータの容量としてもサイズが非常に大きいコンテンツである。漫画という形式に変換することでデータのサイズを大きく削減可能である。

本研究で用いる漫画ならではの要素といえば、静止画でありながら動きを視覚的に表現し、時間の概念を落とし込めるところである。動画の圧縮、容量の削減という点で考

^{†1} 鹿児島大学大学院 理工学研究科
Graduate School of Science and Engineering, Kagoshima University

^{†2} 鹿児島大学 工学部
Faculty of Engineering, Kagoshima University

えた場合、ダイジェストムービー作成や静止画を切り取ってスライドショーのような形にするなどの映像要約技術を用いる手法も考えられる。例えば「野球でバッターがホームランを打った」というシーンであれば、ダイジェストムービーなら打つ前から打った直後までの映像(10秒ほどの動画)になると考えられる。スライドショーなら、打つ前(スイングの最初)と打った瞬間と打った後(スイングの後半)の最低3枚の静止画が必要になると考えられる。漫画であれば、スイングの軌道をスピード線で表すことが可能であり、効果音で「カキーン」等の文字を入れることが出来れば、最低1枚の静止画で表現することができると考えられる。このことにより、自動漫画生成を用いることで、従来の映像要約とは違う形で動画像を圧縮できると思われる。

2. 外部研究状況について

今回の研究は動画像から漫画の生成の研究であるが、他媒体から漫画の生成に関する研究にはいくつかの先行研究がある。

例えば、フィルムコミックの自動生成に関する研究[1]が行われていた。Meng Wangら中国の研究グループによるもので「Movie2Comics」というソフトの開発である。フィルムコミックとは実写やアニメを問わず映画を漫画化したものであり、そのフィルムコミックの自動生成に関する研究である。この研究では顔認識や音声認識を用いて、吹きだしの配置や不要な背景のカットを行う手法の提案がされていた。

また、長谷川誠らによる TVML 台本からの漫画自動生成に関する研究[2]がある。TVML 台本とはテレビ番組をまるまる1本記述することが出来るテキストベースの言語であり、これを独自の TVML プレイヤーに通すことで、あらかじめ用意された3DキャラクターがTVのように原稿どおりに喋るというのが TVML である。この TVML 台本から動画ではなく静止画の形で出力する方法として、漫画の自動生成を行っていた。

他には、首田大仁らによるオンラインゲームのプレイログを用いた漫画の自動生成に関する研究[3]というものもある。プレイログとは「いつ、誰が、どこで、何をして、どうなった」という情報がゲームの記録として保存されているものであり、その情報を用いて漫画を自動生成するシステムを提案していた。

これら関連研究の問題点や改善点として主に次のようなことが挙げられる。動画からシーンセレクトする際に動きの特徴量を用いていないという点、映画や TVML 台本やプレイログといった既に情報を持っているものが素材でなければならないという点、効果線処理などの日本の一般的な漫画に多く見られる漫画的技法に乏しいという点、とい

った3点である。

漫画のコマ割りを自動で行う研究に、高嶋航大らによる物語テキストを基にした漫画のコマの生成手法の提案[4]がある。こちらでも物語テキストという他媒体から漫画を生成する研究であるが、漫画を描く手順である「プロット」「ネーム」「下書き」「仕上げ」の4つの工程のうち、重要な「ネーム」の工程に着目し、コマの配置とコマの内部における構図の決定を行うコマ割りの作業を中心に研究していた。コマの大きさを決める際に重要度を設定し、情報の優劣から重要度を計算して大きさを変化させていた。物語テキストから登場人物の行動や「やがて」などの時間経過を表す単語を基にコマ割りをを行い、漫画を生成していた。

また、漫画の効果線などを動画像の情報から生成する研究に、秋元考夫らによる動的情報を静止画に組み込んだ漫画的表現手法の提案[5]がある。物体の動き情報を抽出し、それらを静止画に表現する漫画的表現手法を提案している。特徴的な動きの推定に SIFT 特徴を用いて、平行移動、回転、スケーリングの各要素を抽出し、それを表現していた。表現はシンプルに矢印を用いて表していた。

3. 提案手法

漫画はそれを取り扱う人や受け手によって定義が若干異なる。そのため、本研究における漫画の定義を以下に示す。以下の3点を満たすものを漫画とする。

- (1)視覚情報として絵があること
- (2)絵と絵に流れが分かるような繋がりがあること
- (3)コマがあり、ページにコマが割ってあること

本節では動画像から自動的に漫画を生成する手法の流れを説明する。

3.1 全体の流れ

提案手法の処理全体の流れを図1に示す。

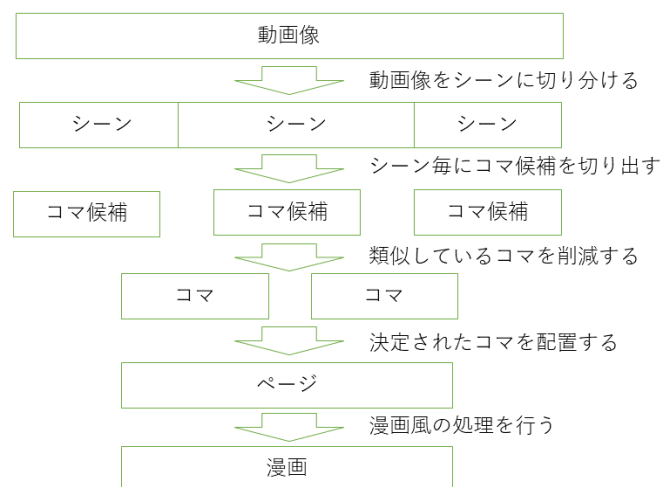


図1 全体の流れ

3.2 シーンの切り分け

カメラの切り替えのない連続したフレームをシーンと定義する。2つのフレーム間のRGB値の誤差 err を以下の(1)式で計算し、誤差が一定値以上あった場合に大きなシーンの切り替わりがあったと判断した。

$$err = \frac{\sum_{i=0}^w \sum_{j=0}^h \sqrt{dr_{ij}^2 + dg_{ij}^2 + db_{ij}^2}}{w \times h} \quad (1)$$

ここで、 dr , dg , db は2つのフレームの各ピクセルのRGB値の差であり、 w は動画の幅、 h は動画の高さである。 i と j は動画中のピクセルの座標を表している。

err の値の変化の推移の例は図2のようになる。このグラフの動画は YouTube に投稿されていた「短編ミステリーホラー 伝言」というタイトルの動画である[6]。

また、ゆっくりと暗転するなどの処理があった際に、切り替わりを検出できないことがあったため、輝度値を利用して暗転・明転を検出した。その際にヒステリシス特性を考慮して閾値を2重に用いることで過剰な検出を抑えている。

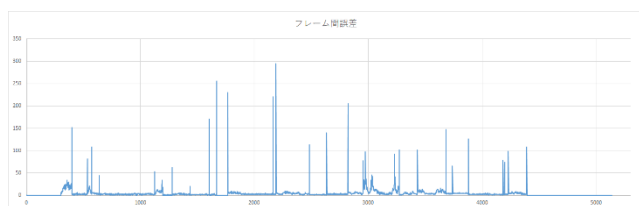


図2 フレーム間誤差の変化

3.3 コマ候補の切り出し

シーンを切り分けした後、シーン毎にコマの候補となるフレームをオプティカルフローと SIFT 特徴量を用いて動画画像から切り出す。オプティカルフローには大きく分けて勾配法とブロックマッチング法があるが、本研究では動画画像に多く含まれがちな拡大・縮小・回転にロバスト性がある勾配法から発展した Farneback 法を用いている。

オプティカルフローを用いて動画中の動きを検出し、動きに変化があった場合、そのフレームを動きの変化点とし、変化点と変化点の間をコマ候補として切り出すことを行う。 f フレーム目と $f-3$ フレーム目でオプティカルフローを計算しており、 f フレーム目に計算した結果をベクトルで表した例が図3である。図2は今回の実験のために撮影した「kuidaore」という動画の1フレームから切り出している。

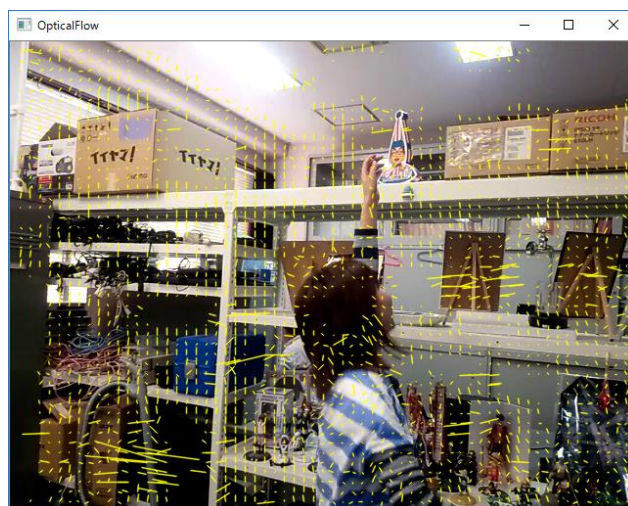


図3 オプティカルフローの計算結果例

ただし、オプティカルフローはカメラ移動の影響を非常に強く受けてしまうため、SIFT 特徴量を用いて動画中の重要な動きを判断する。特徴点検出や特徴量記述のアルゴリズムにはさまざまなものがあるが、回線や照明変動、スケール変化にロバスト性があり、最もポピュラーな SIFT を本研究で用いることにした。

SIFT を利用して動画画像中の f フレーム目と $f-1$ フレーム目の画像を用いて特徴点のマッチングを行う。

マッチングされた点をもとに各特徴点が1フレーム後にどの位置に移動しているかを計算することで特徴点の移動ベクトルを求める。図3と同じフレームでベクトルを赤い線分で表した例が図4である。

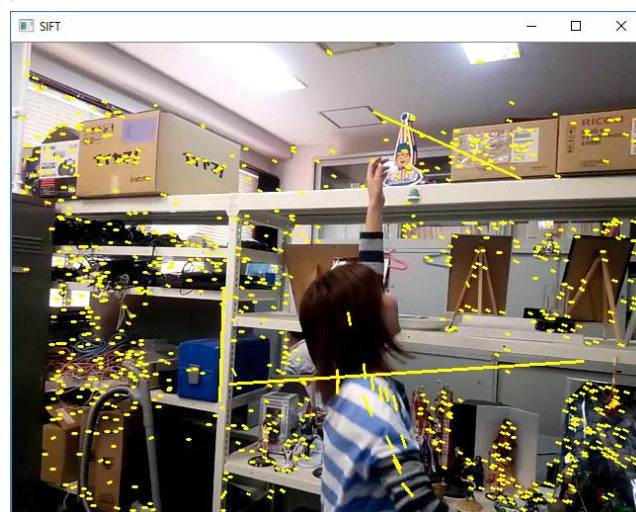


図4 SIFTを用いて特徴点の移動を表示

特徴点の移動を利用して動画全体の動きと違った動きをしている部分を抽出することで、カメラ移動があっても動画中で動いている対象を特定したり、特徴的な動きを捉えることを行う。

まず、動画全体の動きを判断するために、画像全体を $20 \times 15 = 300$ ブロックに分割する。そこから各ブロック内にあるベクトルからブロックがどの向きに移動をしているかを求める。この際、ベクトルの移動量が極端に大きい場合は

計算から除外する．また，ブロック内に複数のベクトルがあった場合，ベクトルを足し合わせてブロック全体のベクトルを決める．図4をブロックに分割したものが図5である．

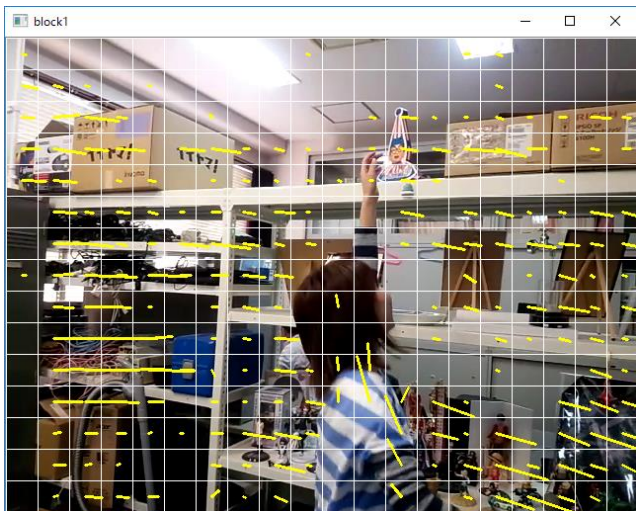


図5 特徴点の動きをブロックで分割

その後ブロックのベクトルを 45 度刻みで分けて，ブロックの移動方向を 8 種類に分類する．8 種類の分類は図5のとおりであり，図6の色のとおりに分類してブロックに色を塗ったものが図7である．

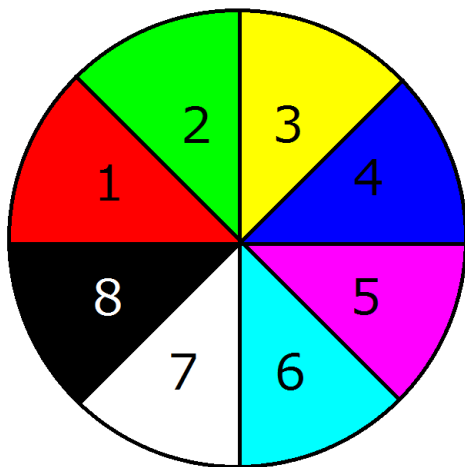


図6 ブロックの移動方向の分類

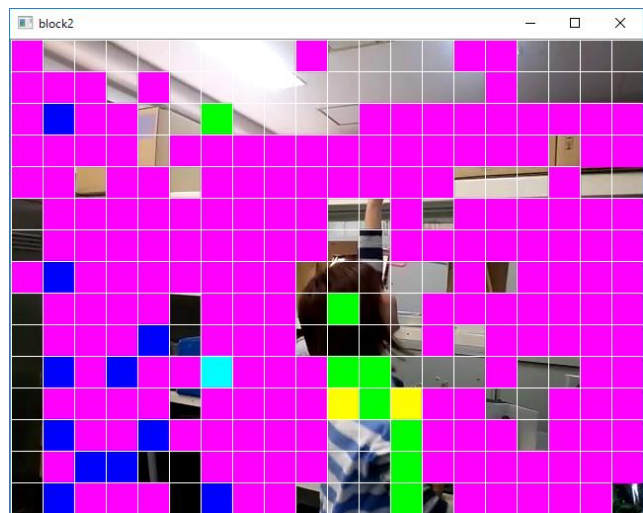


図7 ブロックを移動方向別に色で分ける

動画中の端に映っているものは背景である可能性が高いと考え，動画中の端にあるブロックで最も多い移動方向を背景の動きの向きとして，同じ方向に動いているブロックも背景ブロックとした．また，動きのないブロックも背景ブロックとした．図7の例では，紫色のブロックと色が塗られていないブロックが背景ブロックである．

さらに前後 10 フレームの背景以外の動きも考慮し，各フレームの重要な動きのブロックを判断した．背景以外の動きをした回数が多いほどブロックを赤く塗った結果は図8のようになる．

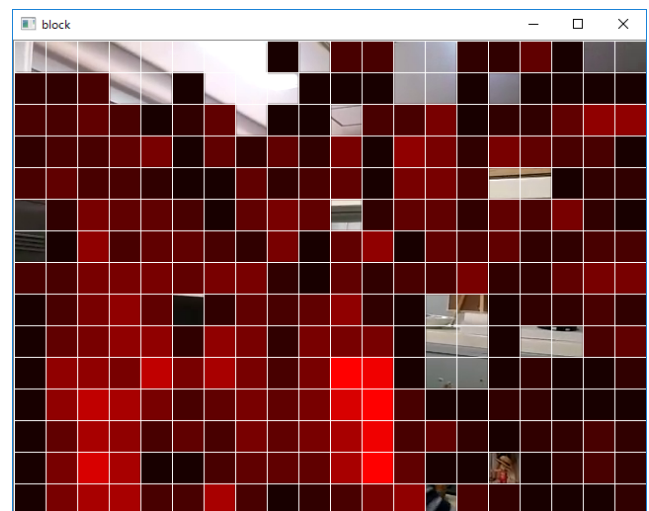


図8 背景以外の動き検出

図8において一定値以上赤いブロックのみオブティカルフローの計算結果を利用して，コマ候補を決定する．

以上の方法で計算した「kuidaore」の動画におけるオブティカルフローの計算結果の推移は図9のようになる．

また，ブロック分割をせずに画面全体でオブティカルフローを計算した場合の推移は図10のようになる．

図9, 図10とも横軸はフレーム数であり，縦軸は移動量の合計である．青がx方向を表しており，赤がy方向を表している．

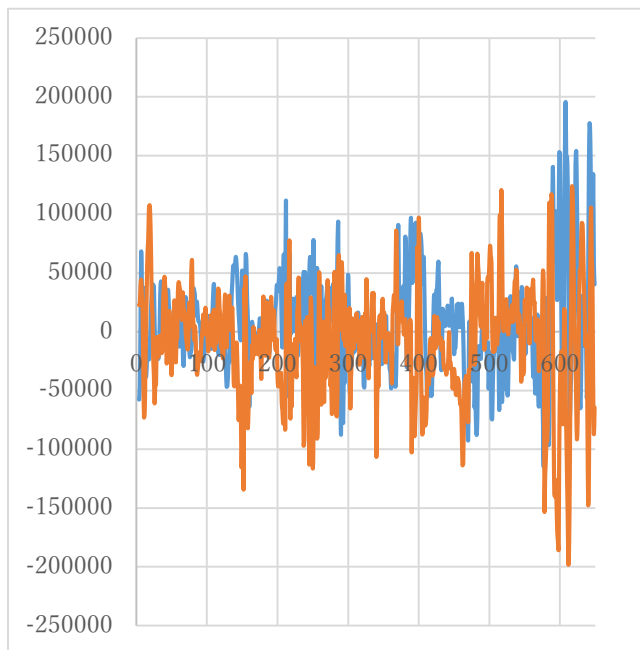


図9 オプティカルフローの推移(SIFT 使用)

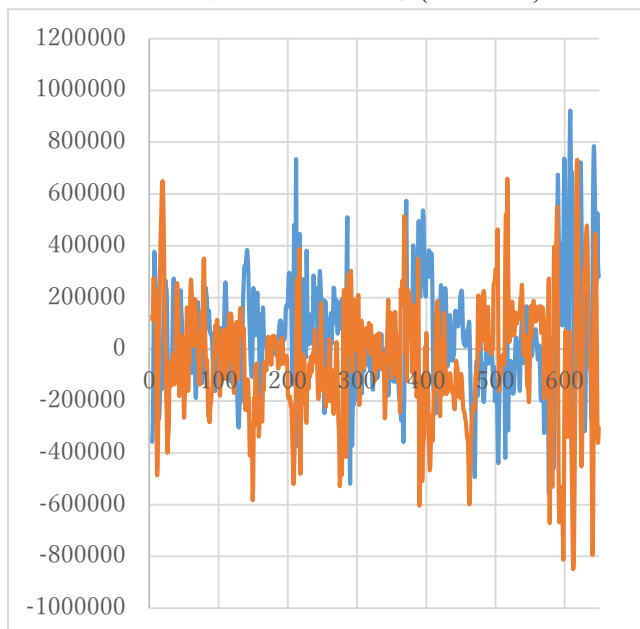


図10 オプティカルフローの推移(SIFT 未使用)

形は似ているところも多いが、とる値の大きさが大きく違う。ブロック分割して利用するオプティカルフローの範囲を限定したほうが、カメラ移動の影響を受けにくく、より細かい動作を計算結果に反映できていると考えられる。

3.4 類似コマの削除

オプティカルフローを利用する関係上、動きが大きい動作が長く続いた場合、同じようなコマが過剰に切り出されることが多く起こる。例えば歩く動作は人からみれば全て「歩く」動作の一連の流れであり、漫画としては1コマか2コマあれば十分であると考えられるが、オプティカルフローによる判断では腕や脚が交互に動く関係上、常にオプティカルフローの変化が激しいので多く切り出される。

そこで形状マッチングを用いて非常に類似しているコマ

を取り除くことにした。形状マッチングとは OpenCv の関数の MatchShapes を用いて2枚の画像の類似度を計算するものである。グレースケールもしくは輪郭だけの2枚の画像に対して3通りの比較を行い類似度を計算する。3つの手法のいずれも7つのHuモーメント不変量からなるHuモーメントを用いている。類似しているほど値は小さくなり、同じ画像であれば0になる。

時間的に連続するコマ候補同士で形状マッチングを用いることで類似しているコマ候補を判断し、時間的に新しい方のコマ候補を削除するようにした。

例えば図11の2枚のフレームは非常に類似しているものと判断して、時間的に新しい下のフレームの画像を削除する。



図11 非常に類似しているコマ候補の例

3.5 コマ割り

各コマには重要度というものを持たせる。重要度はコマ候補切り出しの際に、そのコマのオプティカルフローの計算結果から、動きが大きいコマを重要度が高くなるように1から3の値を持たせているものとする。つまり、動きが大きいコマは重要度3であり、動きが小さいコマは重要度が1であり、その中間の大きさの動きのコマは重要度が2である。

一般的な漫画のコマ割では3段構成であることが多いことから、今回は3段構成のコマ割りのみを考える。1段の重要度が3以上になるように順番にコマを配置していく。また、コマの横幅は各段のコマの重要度の比率に応じて決定する。各段に配置するコマが決定された後、各段の縦幅を、各段のコマの重要度の合計の比率から決定する。

例えば、コマの重要度が順に2, 1, 1, 1, 1, 3, 2, 1, 1, 3, 2, 3, 1, 2...である場合を考える。1ページ目

には重要度が2, 1, 1, 1, 1, 3のコマが配置され, 2ページ目には重要度が2, 1, 1, 3, 2, 3のコマが配置され, 3ページ目には重要度が1, 2...のコマと続いていく.

作成されるページは図12のようになる. 右が1ページ目, 左が2ページ目であり, 各コマの中にそのコマの重要度を書いている. 各ページは右上を1コマ目とし, 左下をページの最後のコマとしている. このあと, 3ページ目以降が続いていく.

また, 各コマとコマの隙間は一般的な漫画の多くに見られる例に従って, 横の隙間を狭く, 縦の隙間をやや広く空けている. これによって, ページの各段を右から左に読みやすくなる.

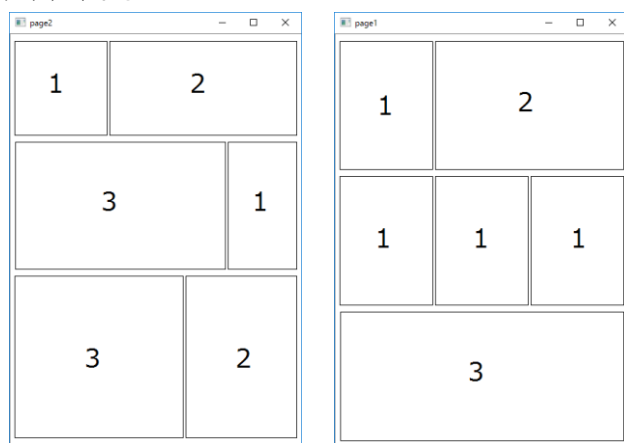


図12 作成コマ割り例

実際に動画をコマ割りをした結果の最初の2ページは図13のようになる. 右が1ページ目であり, 左が2ページ目である.

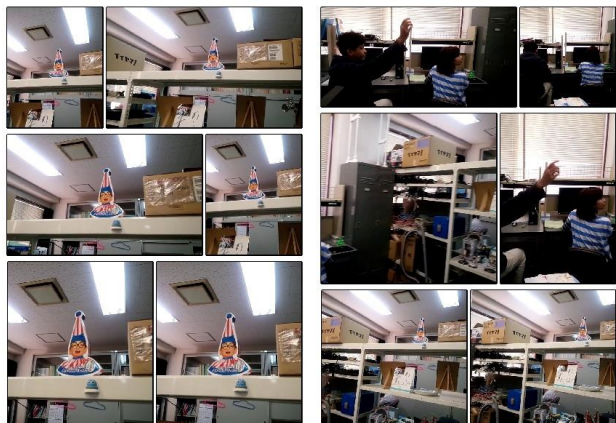


図13 動画を実際にコマ割りした例

3.6 漫画風処理

ページにコマを配置した後, 最後に漫画風の処理を行う. 今回の研究では効果線などの追加を行えなかったため, 簡単な処理のみとなる.

作成されたページの画像に4値化処理を行い, そのうえにエッジ検出で検出した線を重ねる. 図13の2ページを漫画風処理した結果が図14のとおりである.

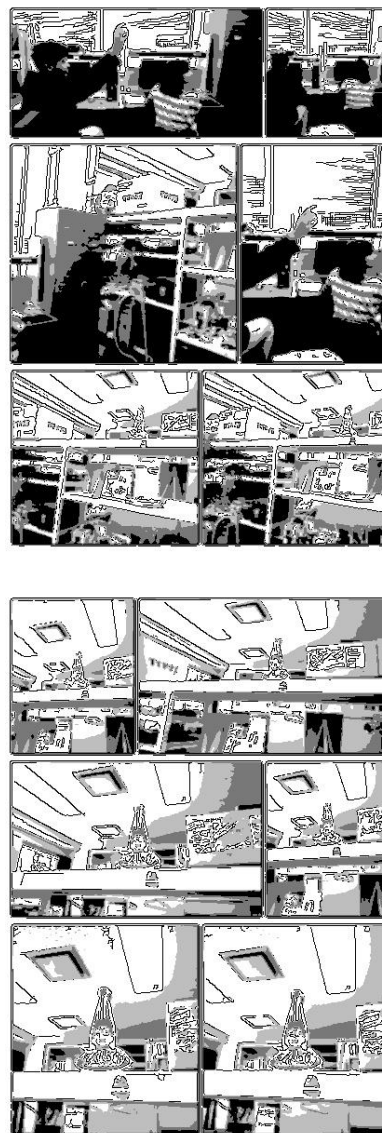


図14 漫画風処理の結果例

4. 評価

完成した漫画の評価をするために, 主に20代の男女に対して2つの動画像でアンケートを行った. 1つ目の動画像は今回の実験で用いている「kuidaore」であり, 2つ目はシーンの切り替わりが多い「dengon」[6]である. アンケートを行った人数は「kuidaore」が11人であり, 「dengon」が7人である.

それぞれの動画で3つの漫画を生成した. 一定間隔で切り出し, 重要度も一定で生成した漫画①, 切り出すフレームと重要度をランダムに決定して生成した漫画②, 今回の研究で自動生成した漫画③の3つである.

アンケートは最初に動画を観てもらい, それから①～③の3つの手法で生成した漫画を読んでもらい, それぞれ質問に答えてもらうことにした. 評価項目は以下の5つである. 5を最もよいものとして1～5の5段階評価をしても

らった。

1. 漫画を読むのにかかった時間
2. 漫画は読みやすかったか
3. 漫画は動画の内容を伝えることが出来ているか
4. 動画の中の大事なシーンがコマに描かれているか
5. 漫画の中のコマの大きさは適切であるか

「kuidaore」は 646 フレーム (21.53 秒) の動画であり、漫画は 42 コマで生成した。①の手法が 7 ページ、②の手法が 8 ページ、③の手法が 7 ページとなった。各漫画を読むのにかかった時間の平均は①が 33.00 秒、②が 31.09 秒、③が 25.45 秒であった。

「dengon」は 5137 フレーム (171.23 秒) の動画であり、漫画は 88 コマで生成した。①の手法が 15 ページ、②の手法が 16 ページ、③の手法が 13 ページとなった。各漫画を読むのにかかった時間の平均は①が 40.86 秒、②が 40.71 秒、③が 39.29 秒であった。

動画像の圧縮という点ではデータ量での削減には成功しているが、「kuidaore」の漫画を読む時間が動画の視聴時間よりも多くかかっているため、短い動画ではよりコマ数やページ数を削減する必要があると考えられる。

「kuidaore」における質問 2～5 の平均点と質問すべての平均点は図 15 のようになった。一番左が全体の平均であり、順に質問 2～5 の結果である。

また、「dengon」におけるアンケートの結果は図 16 である。

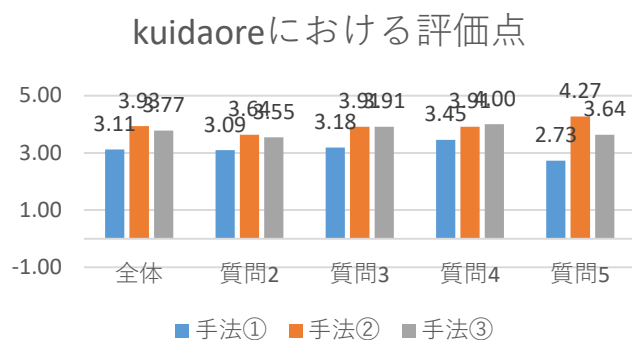


図 15 アンケートの結果 (kuidaore)

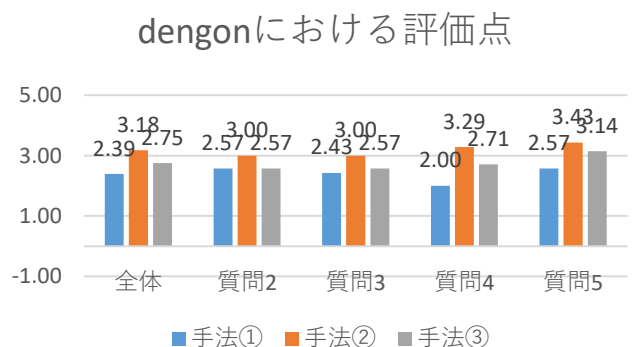


図 16 アンケートの結果 (dengon)

2 つの動画の全ての評価点を平均した値は①2.75 点、②3.56 点、③3.26 点となる。今回のアンケートの結果から、提案手法で自動漫画生成したものは単純に一定時間区切りで切り出して生成したものよりも内容を伝えることが出来ていると考えられる。ただし、ランダムに生成した漫画の方が全体的に評価が高いことから、改善可能な点があると考えられる。

動画像から漫画のコマを切り出すことに関しては、質問 3 と質問 4 の結果から、「kuidaore」の動画に対しては上手くいっていると考えられる。今回の提案手法では主に動きの特徴量を用いてコマ候補を切り出していたため、ホラムービーであり動きがあまりないシーンも多い「dengon」では評価点が低くなったと考えられる。

質問 2 と質問 5 の評価の結果はどちらの動画も低いことから、漫画のコマ割りや漫画風処理には多くの改善点があることが分かる。

5. 結論

本研究では画像処理技術を用いて動画像から自動的に漫画を生成する方法を提案した。

フレーム間差分を計算することと暗転・明転を検出することで、ある程度シーンを大別することに成功し、シーン毎に動画像の中の動きに注目してオプティカルフローと SIFT を用いることで動画中の重要な動きのフレームに着目することでコマ候補を切り出し、コマ候補から形状マッチングを用いてコマを決定し、コマ割りを行うことで最低限の漫画の形を作ることが出来た。

自動生成された漫画は、単純に一定時間区切りで切り出して作成した漫画よりは内容を伝えることに成功していると考えられる。

今回の研究で使った動画以外の動画でも成功するかどうかは不明であり、実験が不十分であるため、現在の手法でどのような動画なら自動漫画生成が可能なかを調べて評価する必要がある。

近年様々な効果処理を施した動画像が出てきており、シーンの大別に用いたフレーム間差分だけでは、対処できないシーンの区切りがある可能性がある。

コマ候補の切り出しは、人物検出を取り入れるなどの工夫で改善が可能であると考えられる。

コマ割りについて、現在は単純に重要度に応じてコマの大きさを変化させているだけであるため、動きに合わせて斜めにコマ割りを行うなどのより漫画の表現を入れることで、内容が分かりやすく読みやすいものになると考えられる。

また、音声認識を用いることでコマの中に人のセリフを入れたり、効果音を表示することが可能になるのではな

いかと考えられる．音声は重要度の計算にも利用可能であると思われる．

参考文献

- [1] MengWang:“Movie2Comics:Towards a Lively Video Content Presentation”, IEEE, vol. 14, no.3, pp.858-870, (2012).
- [2].長谷川誠, 林正樹:”TVML 台本からのマンガ自動生成に関する研究”, 芸術科学会論文誌, vol.1, no.1, pp.15-21, (2002).
- [3].首田大仁, RuckThawonmas”オンラインゲームのプレイログを用いたマンガの自動生成”, ゲーム学会和文論文誌,vol.1,no.1, pp.41-46, (2009).
- [4].高嶋航大:”物語テキストを基にした漫画のコマの生成手法の提案”,研究報告エンタテインメントコンピューティング(EC),vol.13,no.6,pp1-7,(2009).
- [5].秋元考夫,村上和人:”動的情報を静止画に組み込んだ漫画的表現手法の提案”,映像情報メディア学会技術報告,vol.38,no.39,pp1-4,(2014).
- [6].玉大樹:”短編ミステリーホラー 伝言”,Online video.You Tube, <https://www.youtube.com/watch?v=roZLrnb43KQ>, (2014/09/04 に公開)