ヒストグラムの相関に基づく
映像カラー化の参照画像変更

Histogram Correlation based
Reference Image Change
For Video Colorization

令和 3 年(2021)年度卒業論文
指導教員　Micheletto Ruggero

横浜市立大学
国際総合科学部国際総合科学科
物質科学コース

180341
白川雄也

目次

1 序論

1.1 背景

映像情報を用いたビデオ通話等の技術はコミュニケーション手段として重要度が増している. しかし通信インフラが未発達な地域では映像のやり取りが満足にできない場合が考えられる. また高解像度な映像の出現により, 映像データ量は年々増加している. 通信環境が十分でない場所でもより効率的に映像の転送が可能な新しい技術を模索することが必要だ.

近年ディープラーニングは様々な分野で研究され利用されている. 画像や映像の分類・生成, 音声認識や異常検知, さらに楽曲や絵の生成まで及ぶ. その中にはカラリゼーションも含まれる. カラリゼーションとは白黒画像や映像をカラー化する技術である. 色は私たちの脳や心に, 強い印象をもたらすため, 白黒で記録されている古い映画や写真に色をつけ楽しむために利用されることが多い.

カラー化は私たちに深い感動や発見をもたらすだけでなく、データ圧縮に利用できる可能性がある[1][2]. 画像や映像のデータ圧縮手法としてカラー化を利用するには, 送信時に色情報を削減し, 受信側で色情報を復元する手法が考えられる. (Fig 1.1-1)

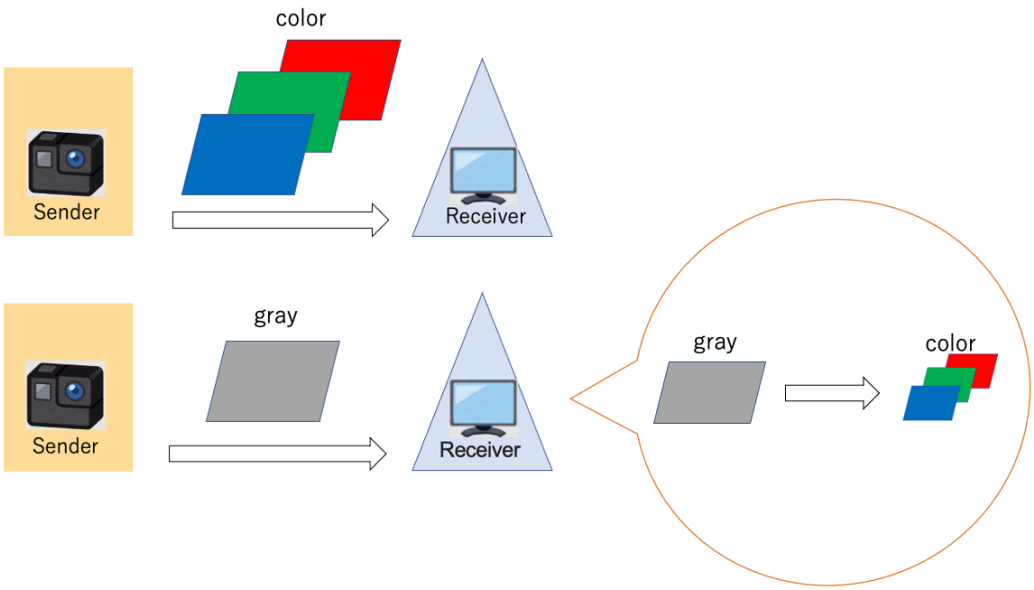


Fig 1.1-1 ： カラー化によるデータ圧縮手法の概略図

1.2 課題と目的

カラリゼーションは多く研究されており,近年はディープラーニングを使用した自動カラー化が多く研究されている[3][4][5].自動で行われるカラー化は色の復元を目的としていないため,非常にリアルな色を作成するが元画像の色との類似性は低いことがあり,その結果,自動着色した結果は,望んだ色でないことがある.例えば,風船のグレースケール画像をカラー化する場合,多くの色が予想される.赤でも良いし,青でも良い.元の風船の色を正確に予測するとは限らない.

データ圧縮手法としてカラリゼーションを活用するには,送信側でカラー画像をグレースケール画像に変化させ,カラー画像より少ないデータ量でグレースケール画像と一部情報のみを送信,受信側ではグレースケール画像と一部情報を用いてカラー画像に復元することが手法として考えられる.映像に適用する場合も同様である.カラー化をエンターテインメントではなく,情報伝達手段として活用するためには復元時には元画像や映像の色と類似していることが前提となってくる.色によって印象や状況が誤って伝わってしまうためである.

また,情報伝達手段として利用するためには,後処理ができないことが制約として加わる.加えて,映像を作成する際にフレーム一枚ずつをユーザーが手動で作業を行うカラリゼーションは労力や時間がかかり適切でない.コミュニケーション手段としてカラリゼーションを利用するには,人の手による作業や後処理を必要とせず,映像のフレームが一枚ずつ出力され,元データの色に類似しているカラー化が自動で行われる必要がある. そして映像をカラー化する際には,時間的な一貫性も考慮しなければならない.一般的に色は時間的に安定しているためだ.

本研究では白黒映像のカラー化を行う手法として Zhang[5]らの手法を用いる.この手法は時間的な色の一貫性を解決し映像のカラー化を行うことが可能であり,参照に用いた画像の色が出力に反映されるカラー化とディープラーニングによる自動カラー化を組み合わせた手法である.本研究では,元の映像フレームを一枚抽出し,参照画像にすることでカラー化後映像の色を復元することを目指す.しかしシーンが切り替わり,参照画像が適切でなくなった場合は自然なカラー映像を得ることができなくなる.もし自然かつ忠実なカラー映像を継続して出力する場合には,映像内の物体や背景,シーンの急激な切り替わりの際に,参照画像も適切に変更する必要がある.フレーム一枚一枚をカラー化するのに元映像のフレームを同じ枚数使用していては意味がないので,参照画像を適切な指針として機能するまで保持し,機能しなくなった場合,新たな参照画像を用いるようにし,色の変化やシーン変化に対応できるようにしたい.参照画像を手動で変更していくのは手間がかかってしまい,さらに参照画像をどのフレームで変更すれば良いか明確な基準もない.そこで本論文では,ヒストグラムの相関を使用し,色の類似で参照画像が適切かどうかを決めることで,参照画像を自動で切り替えることを目指し,品質向上を目指す.

2　関連研究

2.1 畳み込みニューラルネットワーク

一般に畳み込みニューラルネットワークは畳み込み層とプーリング層で構成される. この章では二次元データの形状における畳み込み演算とプーリング処理を説明する.

畳み込み層では, 入力データにフィルター(またはカーネル)を適用し, 畳み込み演算がされる. 入力データにフィルターを一定間隔で動かし、それぞれのフィルターの要素と入力データの対応箇所を乗算し, 和を求め, 出力へ集約していく. このフィルターは最適化されるパラメータを持つ. フィルターの適用する位置をストライドという.

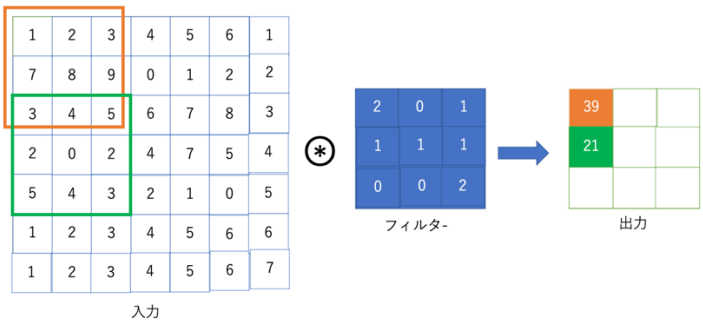


Fig:2.1-1 畳み込み演算

フィルターが出力された後の出力は, 活性化関数に通され, 特徴量マップを得る. なお三次元データに適用する際は, チャンネル方向に畳み込み演算を行った後, 加算して一つの出力を得る.　また入出力を同じサイズにしたり, 出力サイズを調整するため, 畳み込み演算の前に, 入力データの周囲に固定の値を埋めるパディングという処理が行われることがある.

プーリングは空間サイズを小さくする処理を行う. 特徴量のサイズが小さくなるため効率が向上する. プーリングには最大値プーリング, 平均値プーリングがある.

最大値プーリングは、近傍のピクセルから最大値を取得し, 一つの要素に集約する. 平均値プーリングでは, 近傍のピクセルの平均値を取得し一つの要素に集約する. なおプーリング層には最適化されるパラメーターは存在せず, 入出力でデータのチャネル数は変化しない.

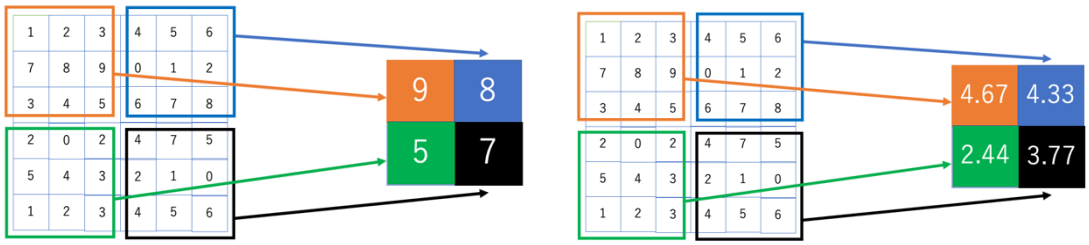


Fig:2.1-2 プーリング処理, 左は最大値プーリング, 右は平均値プーリング(ストライド 3)

CNN は複数の畳み込み層とプーリング層が連続し構成され，最後に全結合層が複数存在する. 全結合への入力が二次元であった場合は, 一次元に変換され入力されることになる. そしてその出力は, 出力層に入力される. 出力層は各クラスの確率を出力する. もし, 1000 種類のクラスに分類する場合は, 1000 個の出力ユニットが存在することになる.

CNN は入力画像から特徴量マップを計算する. 畳み込み層は, 階層的に特徴量を抽出する. 入力層のすぐ後ろにある層はデータから低レベルの特徴量を抽出する. 扱うデータが画像の場合は, エッジ等の低レベルの特徴量を最初のほうの層から抽出する. そして低レベルの特徴量を組み合わせることで, 高レベル特徴量を形成する. 高レベルの特徴量は, 建物や車の全体的な輪郭など複雑な形状を形成することができる. 低レベルの特徴量を層ごとに組み合わせて高レベルの特徴量を形成することで, 特徴量階層を構築する. [6][7]

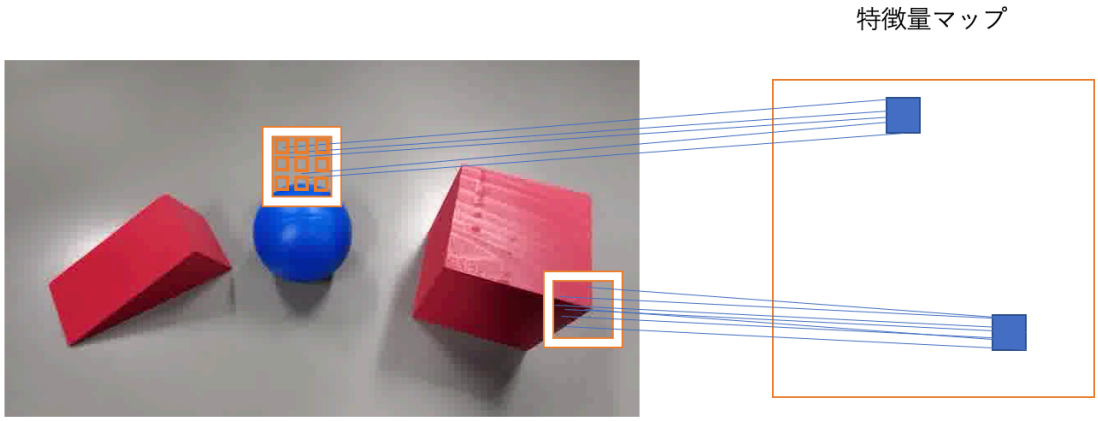


Fig:2.1-3 特徴量マップ算出イメージ

2.2 Very Deep Convolutional Networks for Large-Scale Visual Recognition

VGGNet[8]は畳み込み層とプーリング層からなる畳み込みニューラルネットワークであり，重みを持つ層の数が 11,13,16,19 層のバージョンがそれぞれ存在する．例えば，19 層の VGG は VGG-19 と表されることが多い．VGG は画像分類のためのネットワークで，100 万を超える画像で訓練され，画像を 1000 個のカテゴリに分類することが可能である．また VGG モデルは構造がシンプルなため様々な手法のベースのネットワークや特徴量抽出器として使用されることも多い．

VGG モデルの畳み込み層では 3×3 フィルターを使用し，局所受容野を小さくしている．これは大きいサイズのフィルターを使用するより，小さいサイズのフィルターを重ねて使用することで，提案当時の他の CNN よりも畳み込み層を増加させるためである．畳み込み層のストライドとパディングサイズは 1 に設定されている．また畳み込みの後，活性化関数 ReLU(Rectified linear unit)が存在し，層が重ねられることで非線形による表現力が向上する．畳み込み層での処理を２～３回繰り返した後，最大値プーリング(max-pooling)を行う．最大値プーリングのサイズは 2×2 で、ストライドは 2 に設定されている．最大値プーリングで特徴量のサイズを減らした後，再び畳み込み層が続き，畳み込みが繰り返される．最後に 4096 チャネルの全結合層 2 つと 1000 チャネルの全結合層 1 つが続く．

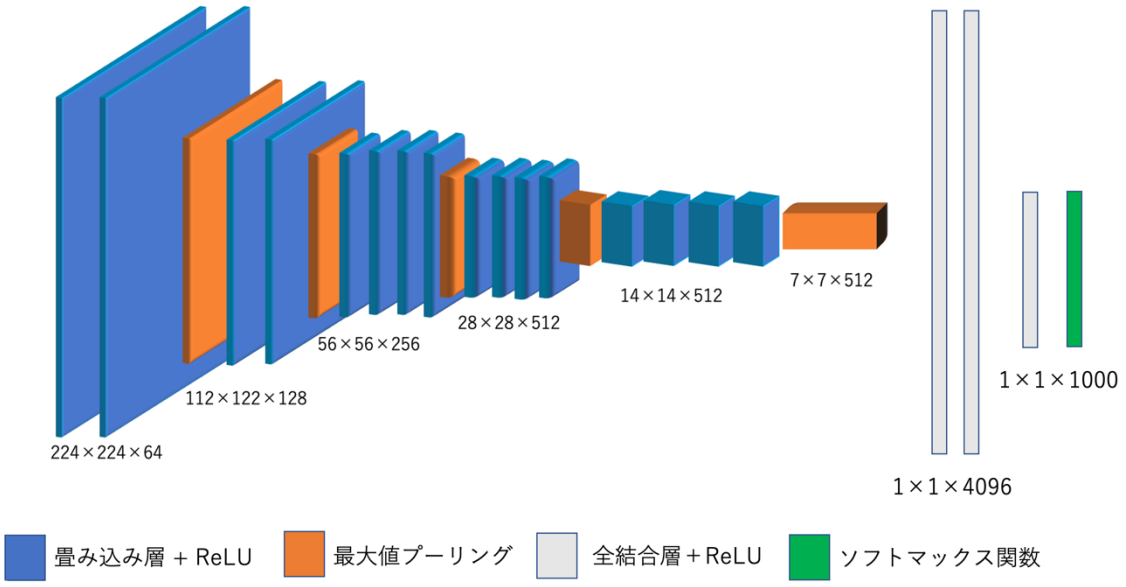


Fig2.2: VGG-19 の全体像

2.3 カラリゼーション

画像や映像をカラー化する技術をカラリゼーションと呼ぶ.現在に至るまで多くの研究がされてきた.Levin ら[9]は最適化手法によるカラー化手法を提案した.この手法は手動によりわずかな色情報をヒントとして与えることでカラー化が行われる.Welsh ら[10]は参照画像の色情報を白黒画像に対応付ける,つまり色を移すことでカラー化を行った.ユーザーが行うのは適切な参照画像を与えることである.適切な参照画像が与えられれば自然なカラー画像が生成される.

また近年,ディープラーニングの研究が盛んにされるようになり,データセットに基づくカラリゼーションが多く研究されている[3][4][5].ディープラーニングによる手法は,カラー画像が多く含まれるデータセットから輝度画像と色差画像の関係を学習することで自動カラー化を行うことができるようになる.

ディープラーニングによる手法の多くは自動的にカラー化を行うが,白黒画像にそれらしい自然な色付けをすることに重きを置かれているため,必ずしも,元の画像や映像の色に忠実なカラー化ではない.データ圧縮に使用するには元の画像や映像の色を忠実に再現する必要がある.

映像のカラー化は,画像よりも困難なものである.それは時間的な一貫性も考慮する必要があるためである.一般的に色は時間的に安定している.例えば,赤い風船は時間が経っても赤いままである.そのため画像のカラー化手法を映像のカラー化にそのまま使用することはできない.映像フレームを個別に着色したとしても,フレーム毎に出力が異なり,色の時間的な一貫性が保たれていない場合がある.カラー化した映像中の物体の色がフレーム毎に変化し,多くのちらつきを感じてしまう.これを改善するためには,隣り合うフレーム間で時間的な隣接関係を構築し,時間的な色の一貫性[11]を保つ必要がある.

2.4 Deep Exemplar-based Video Colorization

本研究では,Zhang ら[5]の手法を利用し,カラー化を行った.この手法のタスクは参照画像に基づく動画のカラー化であり,入力は白黒動画と参照画像である.色の時間的一貫性の問題に対しては,フレームをカラー化した後,その色を次のフレームに伝播させ解決している.しかしこのアプローチは,短い白黒ビデオをカラー化するのに有効であるが,ビデオが長くなるにつれ,エラーが徐々に蓄積し,カラー化が十分でなくなる可能性があるため,一つ前のカラー化後フレームのみを利用するだけでなく,参照として与えられた画像も利用する.与えられた参照画像と白黒フレームとの間にセマンティックな対応関係を見つけ,カラー化の指針にすることで蓄積誤差を低減し,かつ参照画像に忠実なカラー化が可能となっている.

アーキテクチャの簡単な解説を行う.

ネットワークは Correspondence-Subnet と Colorization-Subnet の 2 つのモジュールで構成されている.時刻 t における映像フレームを$x_t^l \in \mathbb{R}^{H\times W\times 1}$, 参照画像を$y^{lab} \in \mathbb{R}^{H\times W\times 3}$とする.$l$ と ab は, それぞれ LAB 色空間における輝度と彩度を表す.

Correspondence-Subnet は,VGG19 から抽出した特徴量マップを用いて,入力フレームx_t^lと参照画像y^{ab}の間のセマンティクな対応関係を構築するネットワークである.

入力白黒フレームが入力された後,それぞれ VGG19 に与えられ,計算される.

最終出力を利用するのではなく,VGG19 内の relu2_2,relu3_2,relu4_2,relu5_2 の層から特徴量マップを抽出し利用する.

抽出した多層の特徴量マップを連結し, x_t^l, y^{ab}に対する特徴量$\Phi x, \Phi y \in \mathbb{R}^{H\times W\times C}$を形成した後,4 つの残差ブロックに供給され,出力は 2 つの特徴ベクトル$F_x, F_y \in \mathbb{R}^{HW\times C}$に再形成される.

次にx_t^l, y^{ab}の特徴間の類似性を計算し,対応関係を見つけるため,位置 i におけるF_xと位置 j におけるF_yの類似性を特徴づける要素を持つ相関行列を計算する。

$$\mathcal{M}(i,j) = \frac{\left((F_x(i) - \mu_{F_x})(F_y(j) - \mu_{F_y})\right)}{\left\|F_x(i) - \mu_{F_x}\right\|_2 \left\|F_y(j) - \mu_{F_y}\right\|_2}$$

ここで,μ_{F_x}, μ_{F_y}は平均特徴ベクトルを表す.

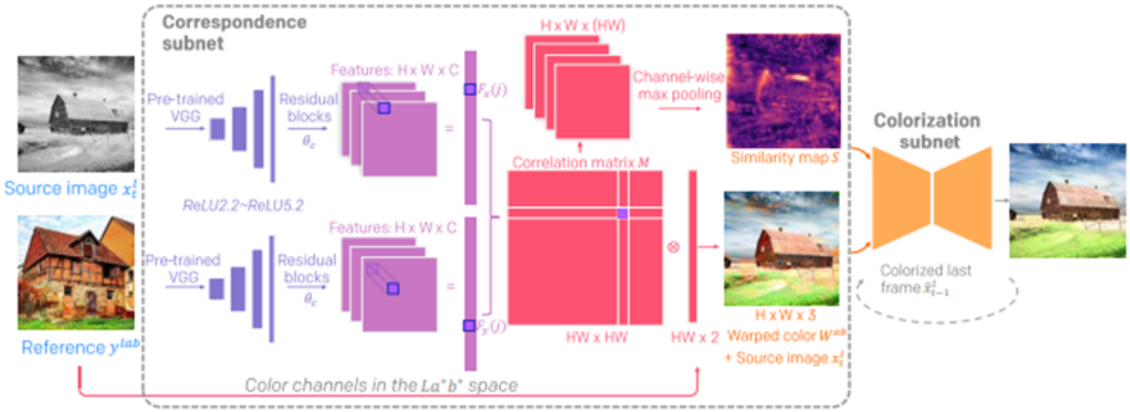
2 つの入力による特徴量を用いた相関行列に従い,参照色y^{ab}をx_t^l方向に転写させ,カラー化の指針となる$\mathcal{W}^{ab}$を得る.

また色の転写は必ずしも正確ではないので,x_t^lの各位置 i ごとに,参照色の抽出の信頼度示す信頼度マップ S を出力する.

$$S(i) = \max_j \mathcal{M}(i,j)$$

よって Correspondence-Subnet は 2 つの出力を生成する.

Colorization-Subnet は現在の白黒フレームをカラー化するネットワークであり，現在のフレームの輝度x_t^l，参照画像から転写されたカラーマップ$\mathcal{W}^{ab}$，信頼度マップ S，前フレームのカラー化結果であるフレーム$\tilde{x}_{t-1}^{lab}$，これら 4 つの入力を受け取る．
出力は t 時点のフレームに対して予測したカラーマップ$\tilde{x}_t^{ab}$である．
Colorization-Subnet は Zhang ら[4]による画像のカラリゼーションで使用された U-Net アーキテクチャに似た構造である．Zhang ら[4]は大規模なデータから学習した情報と，ユーザーが抽出したカラー点を利用することにより，ユーザーの意図を反映するカラー化を実現した．アーキテクチャは低レベルの特徴を再利用するために，スキップ接続を持つオートエンコーダである． 多値の特徴をエンコーダは階層的に捉え，デコーダは利用して現在のフレームの色成分を予測する．ネットワークは複数の畳み込みブロックから構成され，各ブロックは 2〜3 畳み込み層を含む．



(Zhang et al [5])

Fig 2.4-1 : 使用ディープラーニングアーキテクチャのダイアグラム

2.5 カラー化に基づく圧縮

Ftima ら[1]は敵対的生成ネットワークに基づく画像のカラー化を行い,低データレート領域では JPEG より性能が良いことを主観評価で示した.Zhang ら[4]の手法は,グレースケール画像の画素に複数の色情報を与えることで,カラー化を行う.また色情報を与えずとも,自動的に自然なカラー化が可能である,しかし Ftima ら[1]は自動的なカラー化は原画像の色情報との類似性は低いことを指摘し,原画像の色を忠実に再現するため,原画像の画素が持つ色情報を自動的に複数抽出し,拡散させることで原画像の色情報に忠実なカラー化を行った.

また Pan ら[2]は敵対的生成ネットワークを用いたカラー化による映像圧縮を提案した.この手法は H.265 と比べ,より効率的に主観的品質を向上させ,PSNR,SSIM の値を大幅に改善することを示した.

2.6 ヒストグラム

本研究ではヒストグラムを用いてカラー化時の参照画像を適宜変更するため,ヒストグラムの解説をする[12].

横軸に画素値,縦軸にそれぞれの画素値の頻度(画素値をもつ画像の個数)をとり,画像の画素値の分布を棒グラフで表したものを,ヒストグラムとよぶ.ヒストグラムは,画像中にどのような値の画素値がどれほど含まれているかを分布として示す.カラー画像の各色チャネルに対しても,グレースケール画像と同様にヒストグラムを求めることができ,各色チャンネルのヒストグラムには,画像に含まれる色彩に関する特徴が表れる.また画像中の物体位置とは関係ない.

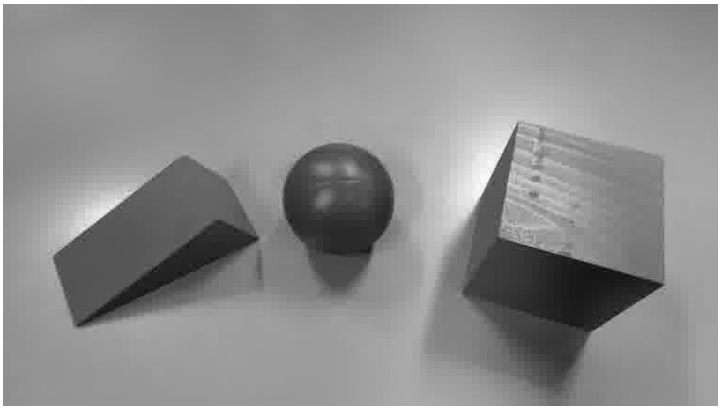


Fig2.6-1: グレースケール画像

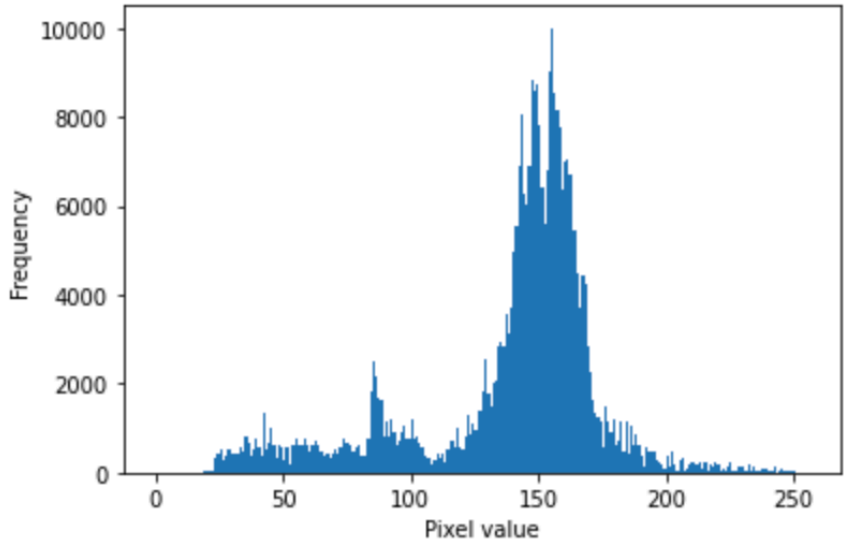


Fig2.6-2: ヒストグラム

2.7 Histogram Correlation for Video Scene Change Detection

グレースケールのヒストグラムを用いて,映像のシーンの切り替わりを検出する Radwan ら[13]の手法が存在する.Radwan は,フレーム間の相関に基づくシーン検知アルゴリズムを提案した.

このアルゴリズムは映像の最初のフレームを参照フレームとし,参照フレームのヒストグラムと全ビデオフレームのヒストグラムの相関を計算した後,計算された相関値とフレーム数の関係をプロットし観察することで,シーンと動きの変化を検知し区別する手法である.

Radwan らは,あるフレーム数で相関値が一定の場合,参照フレームと比べ,背景が変化していない動きのあるシーンであると示した.さらに,フレーム数に応じて相関値が変化している場合は,シーンが徐々に変化していることを示し,相関値が急激に変化する場合は,シーンが急激に変化していることを示した.

まず最初のフレームを参照フレーム fr とし,このときのヒストグラム G_{fr} は,

$$G_{fr}(r_k) = n_k$$

ここで, r_k は K 番目の濃度値, n_k はフレーム内で濃度値が r_k である画素の数である.

フレームが複数枚ある映像のヒストグラム $G_{t\,(t=2,3\cdots)}$ を計算し, G_{fr} と G_t の相関は次のように計算できる.

$$corr(G_{fr}, G_t) = \frac{\sum_{u=1}^{n}\left((G_{fr}(u) - g_{fr})(G_t(u) - g_t)\right)}{\sqrt{\sum_{u=1}^{n}(G_{fr}(u) - g_{fr})^2 \sum_{u=1}^{n}(G_t(u) - g_t)^2}}$$

ここで, n はグレースケールの濃度値 r_k の数, g_{fr} と g_t はそれぞれ G_{fr} と G_t の平均値である.

アルゴリズムステップは以下の通りである.

1. 映像をフレームに分割.
2. 最初のフレームを参照フレームとし,そのヒストグラム G_{fr} を計算.
3. 映像の残りフレームについて,ヒストグラム G_t を計算する.
4. G_t と G_{fr} の相関を計算する.
5. シーンや動きの変化を推測するために計算した相関とフレーム番号の関係をグラフにプロット.

3　本研究の手法

3.1 ヒストグラムによる参照画像自動切り替え

Radwan[13]らの,映像のフレーム間の相関に基づく,シーンチェンジ検出アルゴリズムを拡張し利用することで,カラー化時の入力である参照画像を適宜変更する.

映像のカラー化に使用する Deep learning アーキテクチャは Zhang ら[5]の学習済みモデルを用いた.Zhang らによる先行研究ではカラー化時に用いる参照画像は不変であるが,本研究では参照画像と現在のフレームのヒストグラムの相関値が閾値を下回った際に,参照画像を変更し,現在のフレームを新たな参照画像にする.再びヒストグラムの相関値による参照画像が切り替わるまで,カラー化には新たな参照画像を使用する.

また Radwan らの手法ではヒストグラムを算出する際,映像フレームのグレースケール画像に適用していたが,本研究では各カラーチャネル RGB にそれぞれに適用した.またヒストグラムを算出する際の参照フレームもカラー化参照画像と同じタイミングで切り替えを行う.

閾値は-0.9 から 0.9 の範囲で実行し,切り替わりやすさを変更.切り替え数による品質の関係を確認した.

拡張後のアルゴリズムのステップ(6 は,相関値が閾値を下回るかどうかで A,B 別れる.)

1. ビデオをフレームに分割し,以後それぞれのフレームを一枚ずつ処理していく.
2. 第一枚目フレームをカラー化の参照画像 y,ヒストグラムの参照フレーム fr とする.
3. 参照フレーム fr のヒストグラム G_{fr} が算出されていない場合は算出.
4. 時刻 t におけるフレーム x のヒストグラム G_t を色(RGB)ごとにそれぞれ算出.
5. 参照画像のヒストグラム G_{fr} と時刻 t におけるフレームのヒストグラム G_t の相関を RGB ごとに計算する.

6.A もし,相関値が事前に設定した閾値を下回った場合,時刻 t におけるフレームと参照画像は色の類似度が下がったと考え,時刻 t におけるフレーム x を新たな参照画像 y と参照フレーム fr として置き換える.(次に参照画像が切り替わるまで,グレースケール画像をカラー化する際や相関を計算する際はこの参照画像を使用する.)この場合,時刻 t におけるフレーム x のグレースケール化及びカラー化は行わず,カラー化後のフレーム $\tilde{x}$ として扱う.次のフレーム($t+1$)の処理のため,ステップ 3 に進む.

6.B もし閾値を下回らない場合は,参照画像や参照フレームの変更は行わない.(時刻 t におけるフレーム x のグレースケール画像のカラー化や,時刻 t+1 におけるヒストグラム G_{t+1} の相関算出の際には,以前の参照画像が引き続き使用され実行される.)
時刻 t におけるフレーム x がグレースケール画像に変換され,カラー化される.
次のフレーム($t+1$)の処理のため,ステップ 4 に進む.

3.2　評価指標

3.2.1 Peak Signal-to-Noise Ratio (PSNR)

評価指標として Peak Signal-to-Noise Ratio(PSNR)を用いた.PSNR は画像の客観画質評価の指標としてよく利用され,画素値の平均二乗誤差(MSE：mean square error)を比較することで算出される.

本研究では時刻 t におけるカラー化後フレーム$\tilde{x}$と元映像のカラーフレーム x を使用し,PSNR を算出.全てのフレームに適用した後,平均を算出し、映像の PSNR とした.

$$PSNR = 10\left(\frac{max^2}{\frac{1}{HW}\sum_{i=1}^{H}\quad\sum_{j=1}^{W}\quad\left(\tilde{x}_{i,j}-x_{i,j}\right)^2}\right)$$

ここで max は最大ピクセル値 255 であり,$x_{i,j}$は $H\times W$の原画像の i,j番目の画素値,$\tilde{x}_{i,j}$は比較画像の i,j番目の画素値である.[14]

3.2.2 Structural Similarity index (SSIM)

また Structural Similarity index (SSIM)による評価も行う. SSIM は明暗,コントラストを分離した画像の類似度を評価する.[14]

$$SSIM = \frac{\sigma_{x,\tilde{x}}}{\sigma_x \sigma_{\tilde{x}}} \cdot \frac{2\sigma_x \sigma_{\tilde{x}}}{(\sigma_x)^2 + (\sigma_{\tilde{x}})^2} \cdot \frac{2\overline{x}\overline{\tilde{x}}}{(\overline{x})^2 + (\overline{\tilde{x}})^2}$$

ここで$\overline{x}$は原画像の画素値平均,$\overline{\tilde{x}}$は再現画像の画素値平均, $\sigma_x, \sigma_{\tilde{x}}$はそれぞれの標準偏差, $\sigma_{s,y}$は両者の相互共分散である.第 1 項は,画素間の相関の強さを表し,第 2 項は,標準偏差の差分の大きさを評価し,コントラストの変化を評価する値となる.第 3 項は明暗の変化を評価する値である.

3.3 検証映像データ

本研究の検証データとして,筆者が撮影した映像,Pixabay[15]より取得した[16][17][18]を利用する.なお取得した映像はPixabay Licenseで非商用,商用共に無料で帰属表示も必要がない映像である.全ての映像サイズは(432×768×3)である.

Table 3.3-1 ： 筆者が撮影したカラー映像

映像番号	映像内容
映像①	机の表面が映された後、赤の正六面体と三角柱、青の球が現れる
映像②	横浜市立大学理学系研究棟の外を歩いている
映像③	公園を歩いている内容.ベンチや遊具が映る

Table 3.3-2: Pixabayより取得したカラー映像

映像番号	映像内容
映像④[16]	ローラースケートで滑っている男性が映る
映像⑤[17]	定点映像である.道路が映されており,車や路面電車が通る
映像⑥[18]	定点映像である.歩道,車道が映されており,歩行者や車が通る

4 結果

4.1 検証映像①の出力結果

Table 4.1-1:検証映像①における設定閾値とそれに伴う参照画像切り替え数,PSNR,SSIM

閾値	切り替え数	PSNR[dB]	SSIM
-0.9 から 0.5	0	21.830	0.930
0.6	1	27.839	0.923
0.7	1	27.791	0.922
0.8	3	29.180	0.922
0.9	5	29.782	0.923

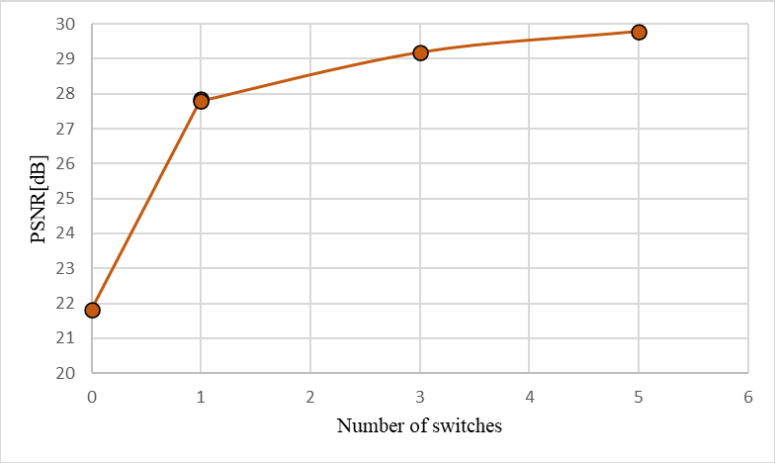


Fig4.1-1: 参照画像の切り替え数と PSNR の関係

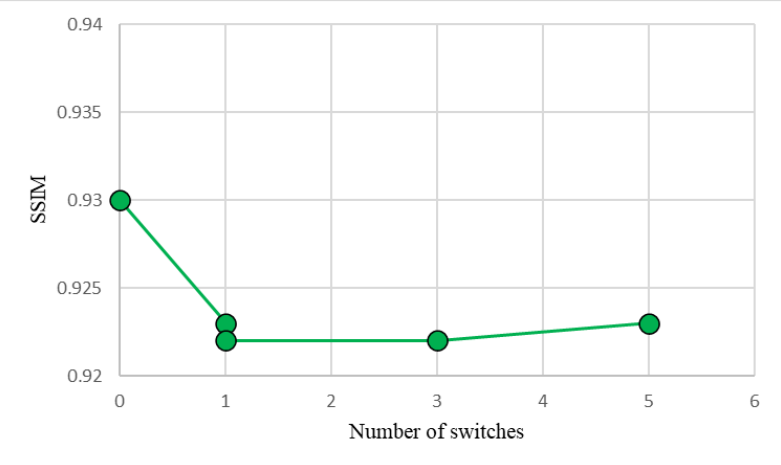


Fig4.1-2:参照画像に切り替え数と SSIM の関係

Fig4.1-3 検証映像①における出力映像フレーム

Fig4.1-4 検証映像①における出力映像フレーム

4.2 検証映像②の出力結果

Table4.2-1 検証映像②における設定閾値とそれに伴う参照画像切り替え数,PSNR,SSIM

閾値	切り替え数	PSNR［dB］	SSIM
-0.9 から 0.1	0	23.002	0.923
0.2	2	24.526	0.925
0.3	3	25.022	0.927
0.4	4	25.193	0.927
0.5	8	26.791	0.929
0.6	15	27.800	0.931
0.7	20	27.782	0.931
0.8	35	28.239	0.932
0.9	63	28.454	0.933

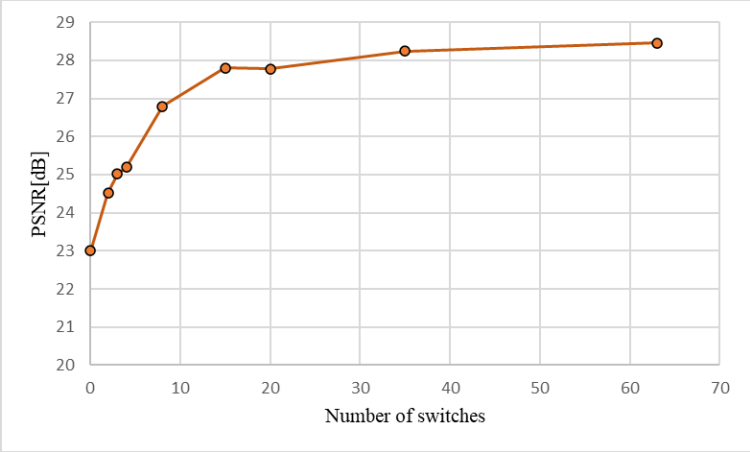


Fig4.2-1 参照画像の切り替え数と PSNR の関係

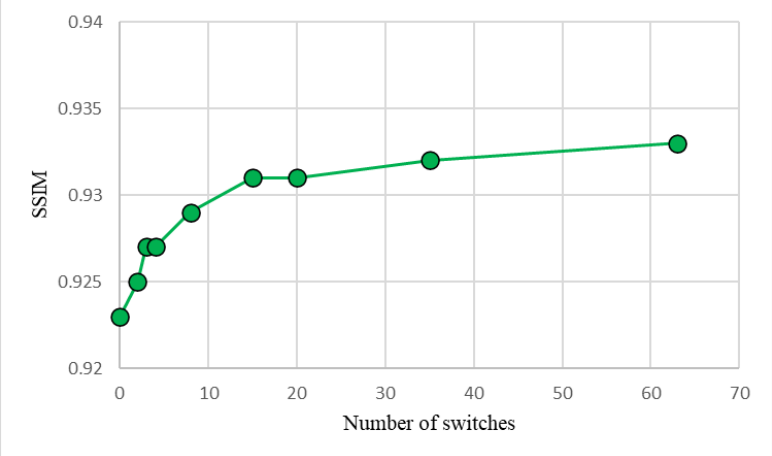


Fig4.2-2 参照画像の切り替え数と SSIM の関係

Fig4.2-3: 検証映像②における出力映像フレーム

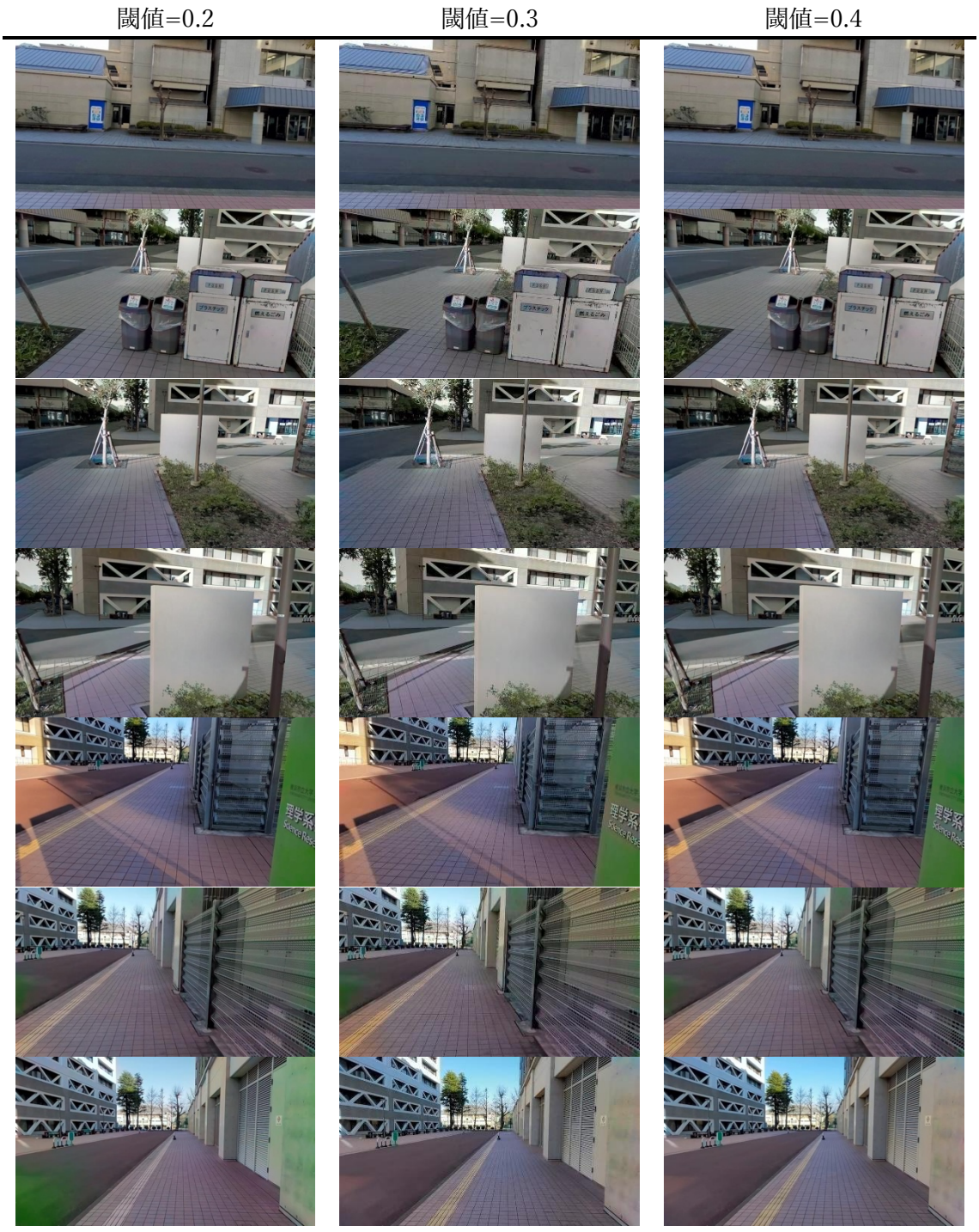


Fig4.2-4: 検証映像②における出力映像フレーム

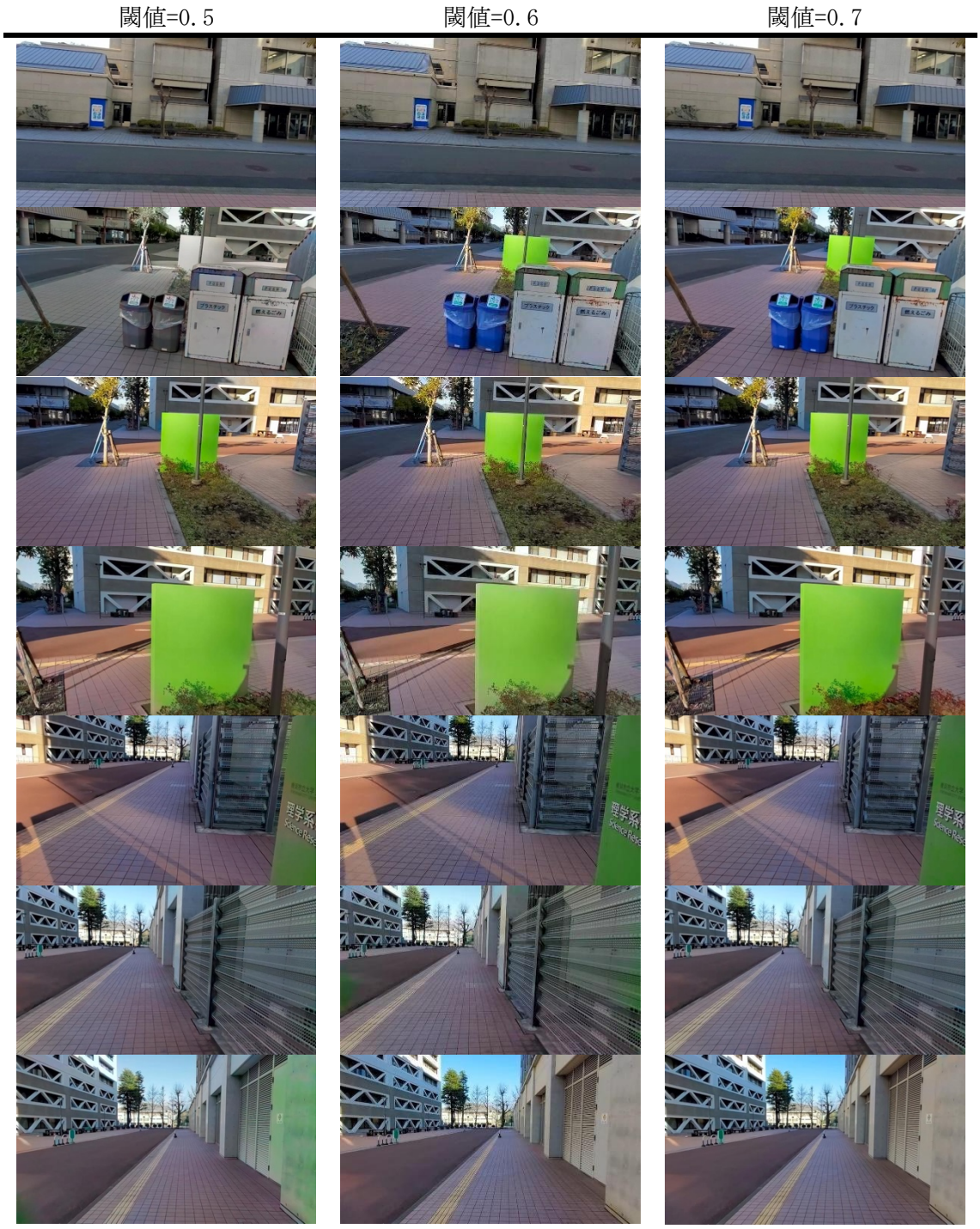


Fig4.2-5: 検証映像②における出力映像フレーム

Fig4.2-6: 検証映像②における出力映像フレーム

4.3 検証映像③の出力結果

Table 4.3-1:検証映像③における設定閾値とそれに伴う参照画像切り替え数,PSNR,SSIM

閾値	切り替え回数	PSNR［dB］	SSIM
-0.9 から -0.4	0	21.639	0.911
-0.3	2	22.615	0.914
-0.2	6	22.858	0.917
-0.1	7	24.298	0.920
0.0	6	24.254	0.921
0.1	10	24.735	0.921
0.2	15	26.107	0.930
0.3	22	27.497	0.930
0.4	25	27.698	0.931
0.5	29	28.100	0.931
0.6	40	28.349	0.932
0.7	51	28.796	0.933
0.8	73	29.058	0.934
0.9	112	29.413	0.934

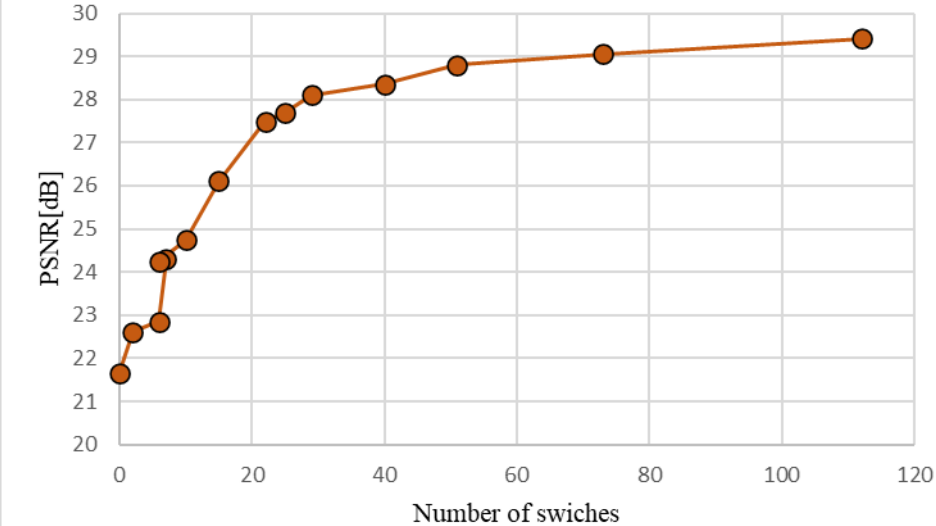


Fig 4.3-1 参照画像切り替え数と PSNR の関係

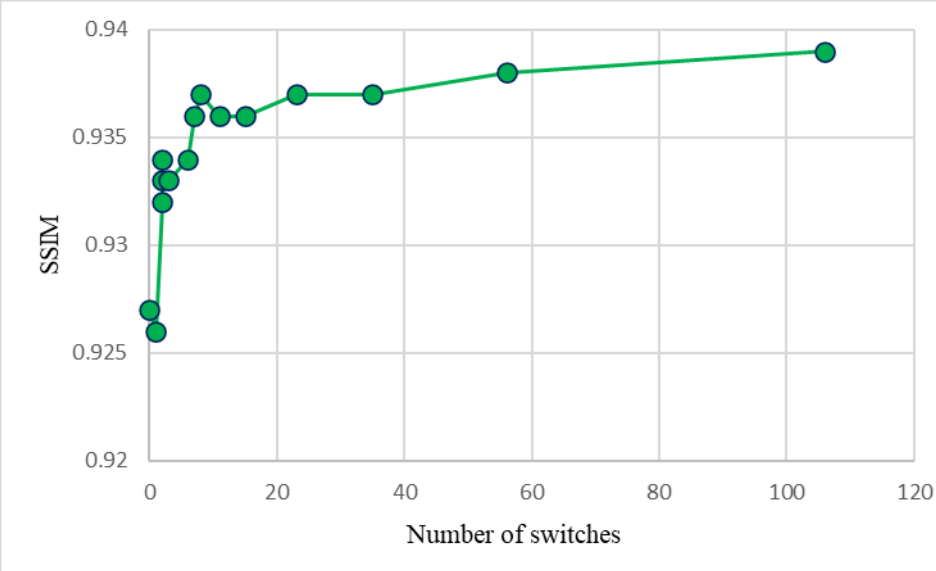


Fig 4.3-2 参照画像の切り替え数と SSIM の関係

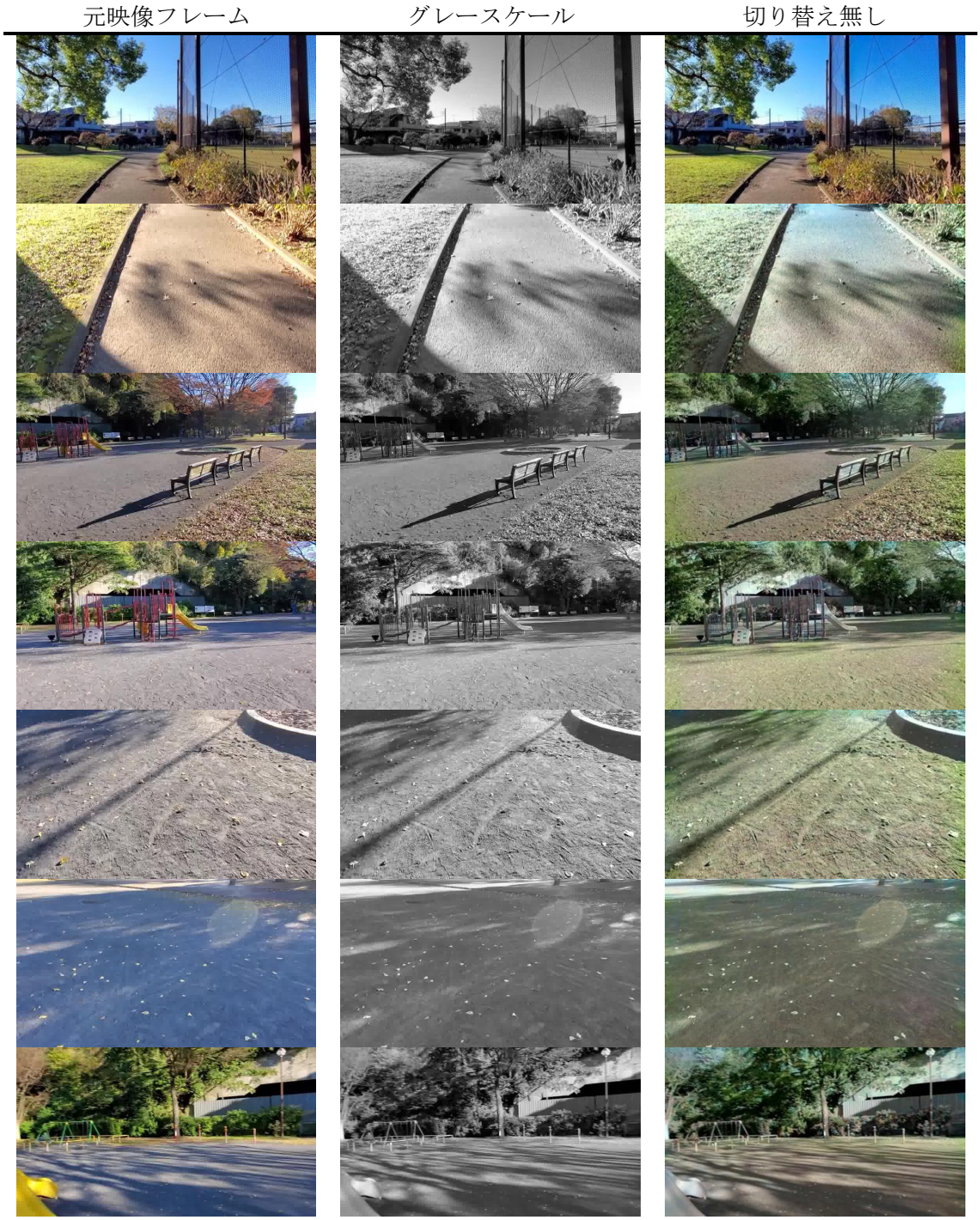


Fig4.3-3: 検証映像③における出力映像フレーム

Fig4.3-4: 検証映像③における出力映像フレーム

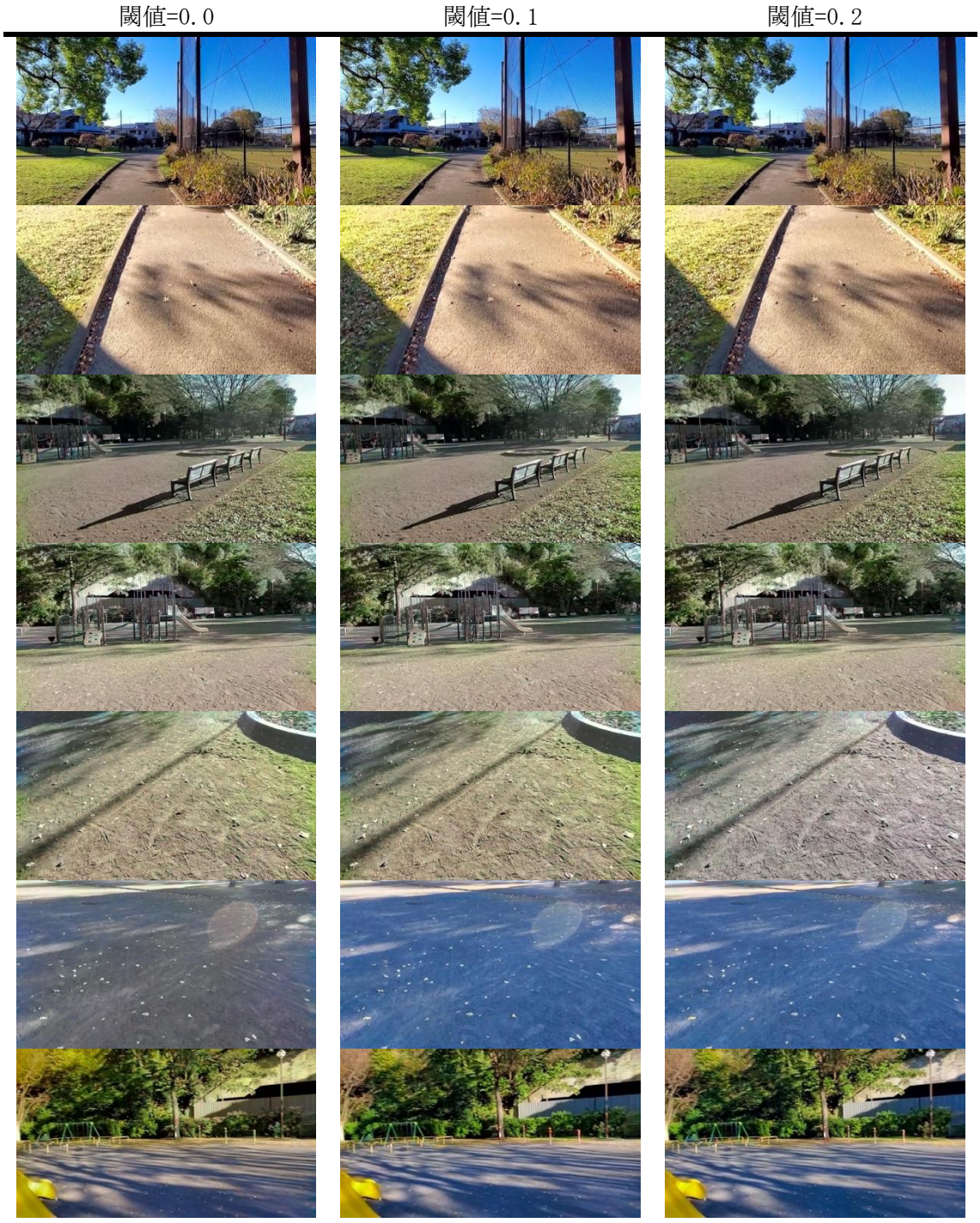


Fig4.3-5: 検証映像③における出力映像フレーム

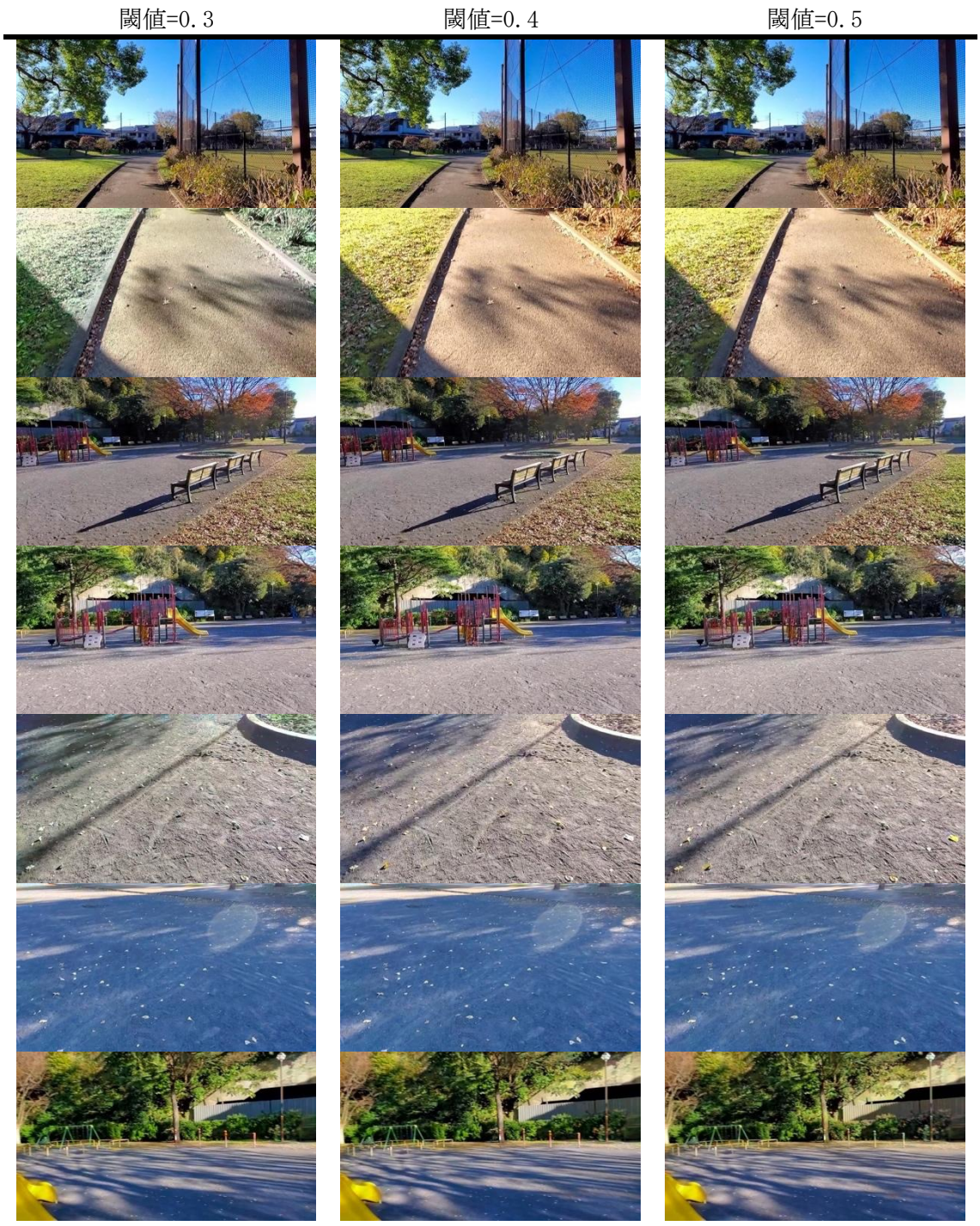


Fig4.3-6: 検証映像③における出力映像フレーム

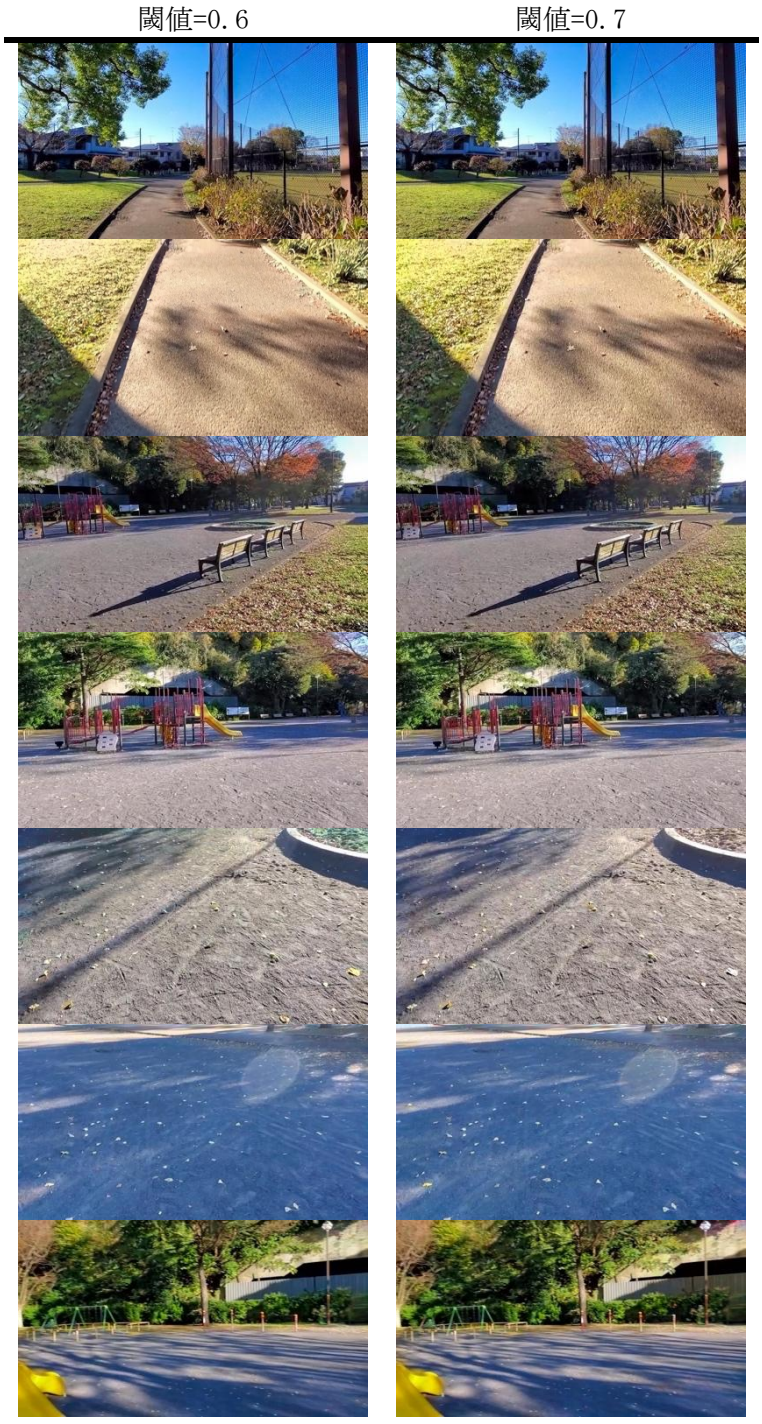


Fig4.3-7:検証映像③における出力映像フレーム

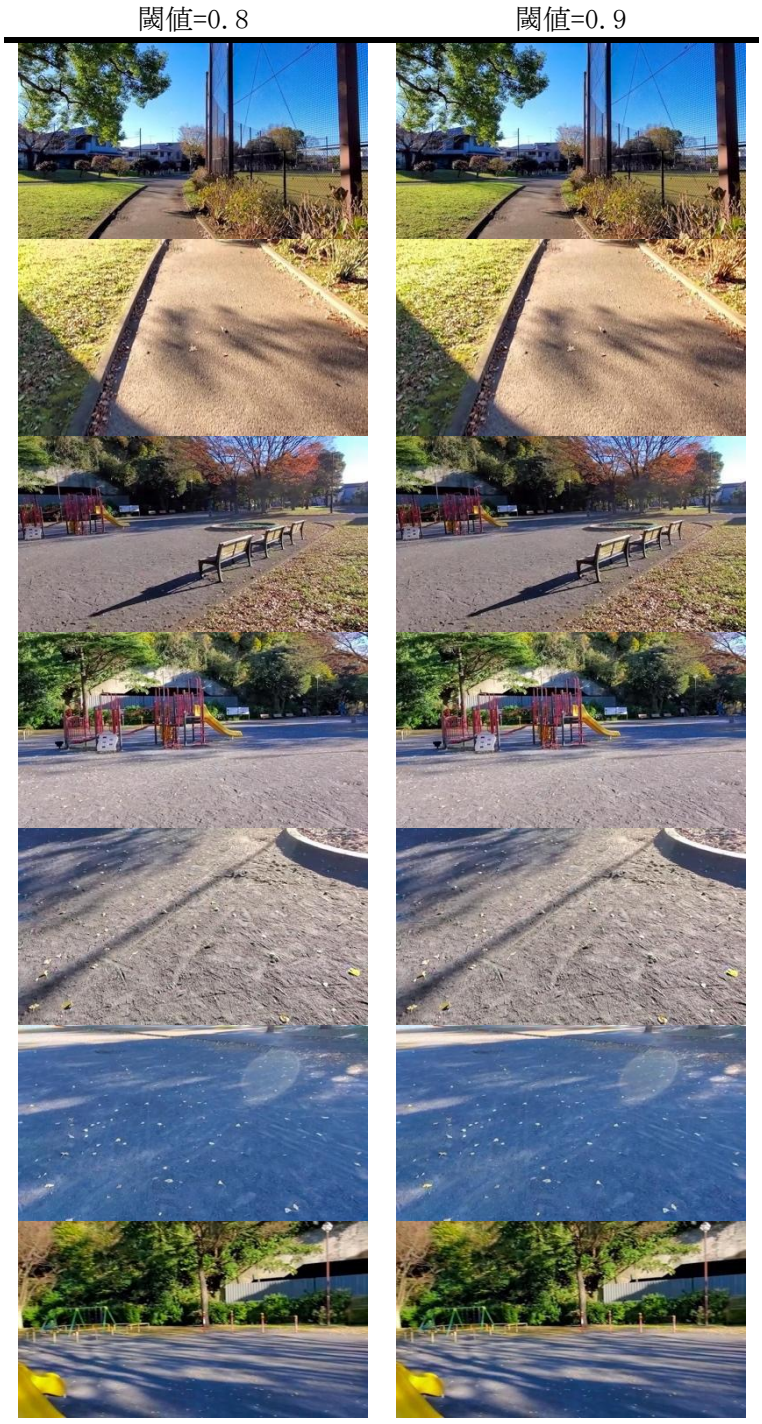


Fig4.3-8: 検証映像③における出力映像フレーム

4.4 検証映像④の出力結果

Table 4.4-1 ： 検証映像④における設定閾値とそれに伴う参照画像切り替え数,PSNR,SSIM

閾値	切り替え数	PSNR［dB］	SSIM
-0.9から-0.5	0	27.952	0.927
-0.4	1	28.262	0.926
-0.3	2	28.869	0.932
-0.2	2	30.042	0.934
-0.1	2	29.953	0.933
0.0	3	29.793	0.933
0.1	6	29.935	0.934
0.2	7	29.947	0.936
0.3	8	30.196	0.937
0.4	11	30.236	0.936
0.5	15	30.260	0.936
0.6	23	30.217	0.937
0.7	35	30.517	0.937
0.8	56	30.670	0.938
0.9	106	30.985	0.939

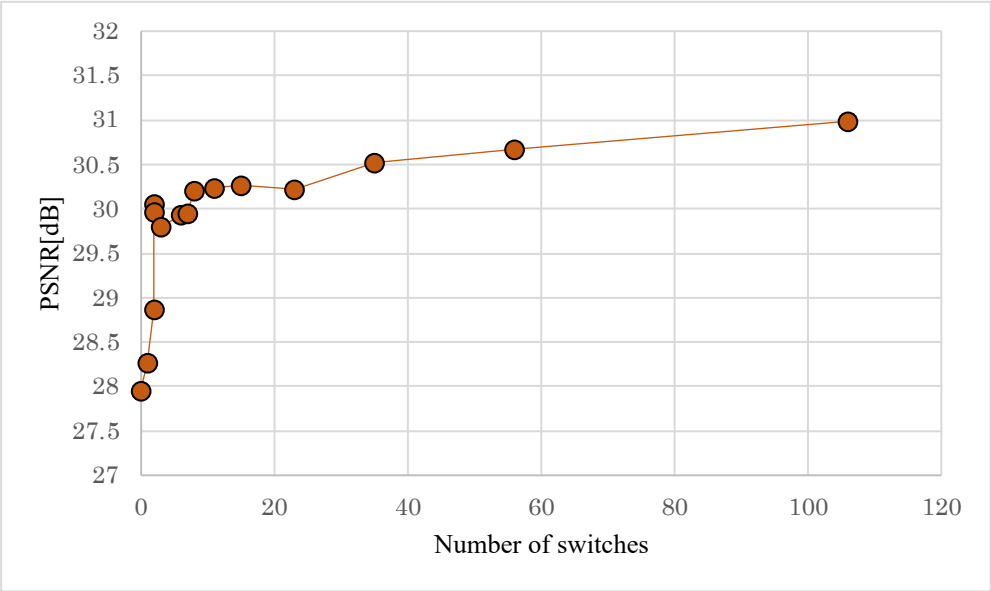


Fig 4.4-1 ： 参照画像の切り替え数と PSNR の関係

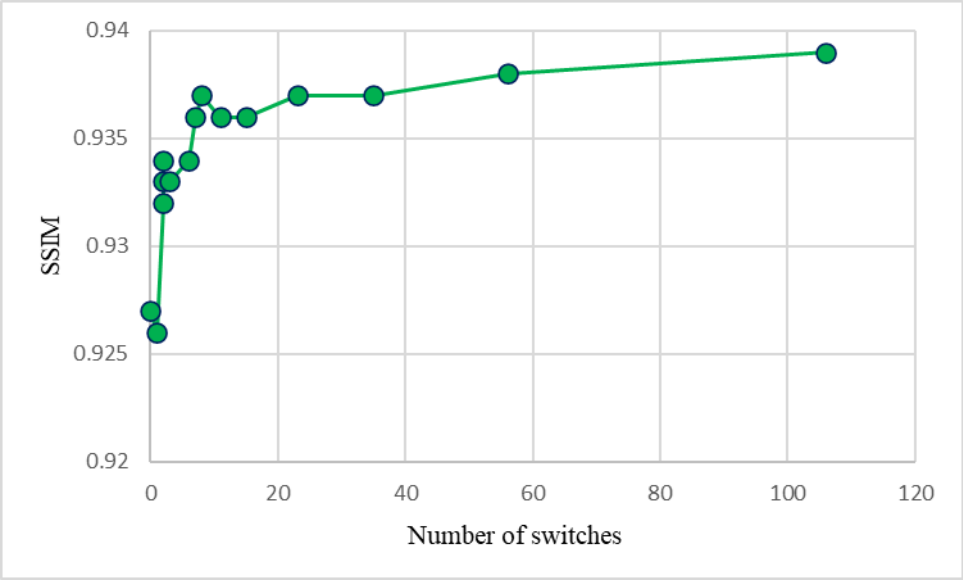


Fig 4.4-2 参照画像の切り替え数と SSIM の関係

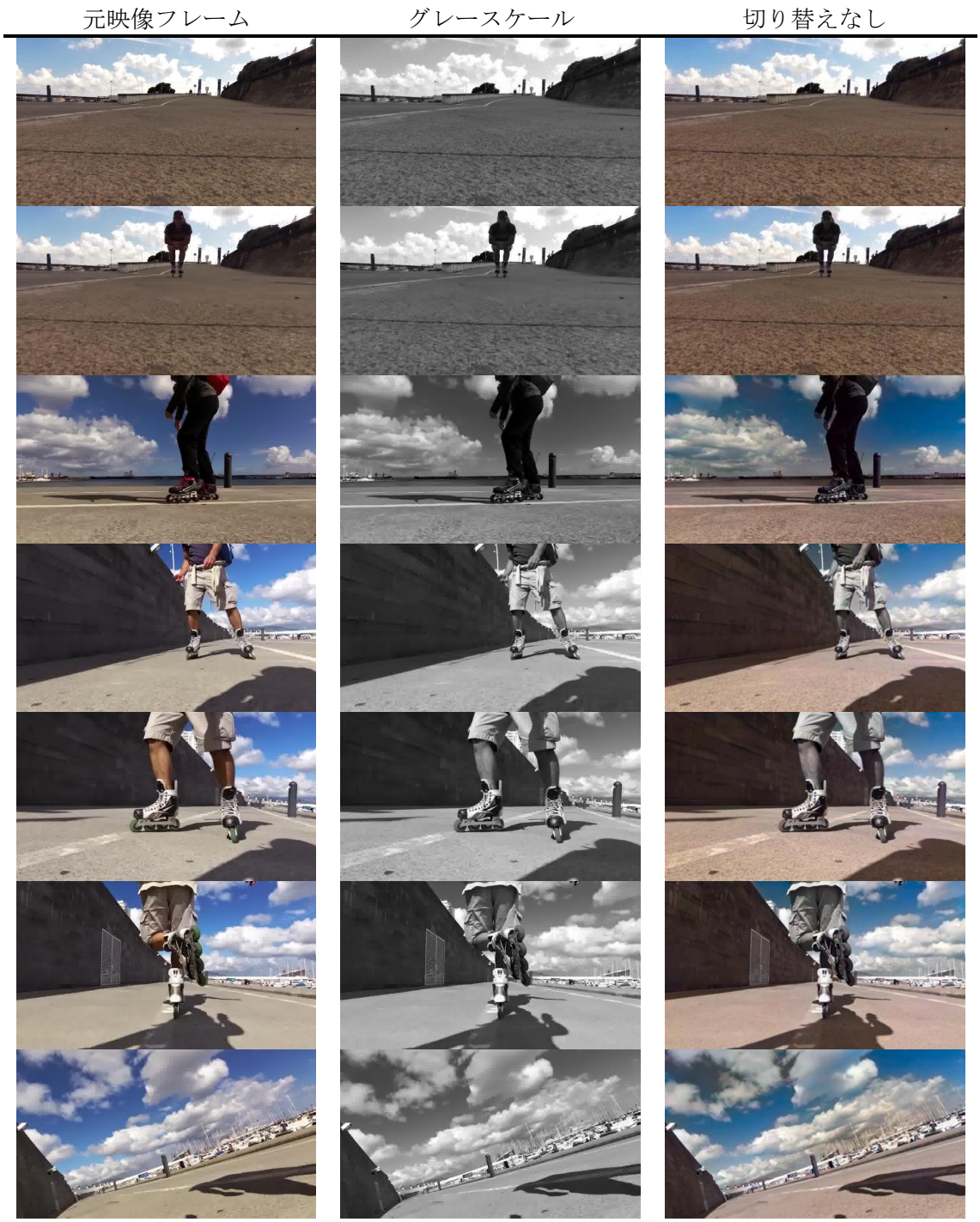


Fig 4.4-3 検証映像④における出力映像フレーム

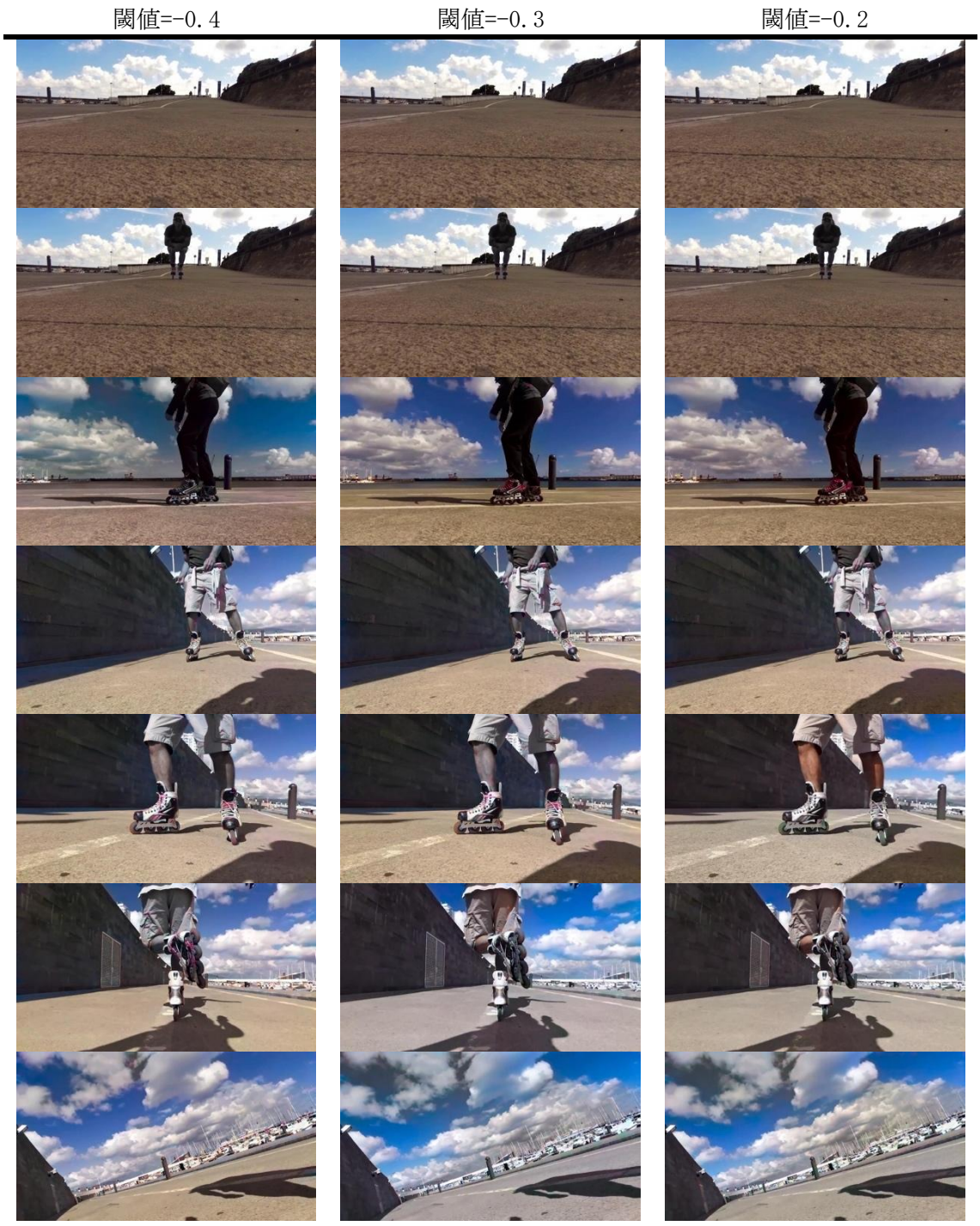


Fig 4.4-4 検証映像④における出力映像フレーム

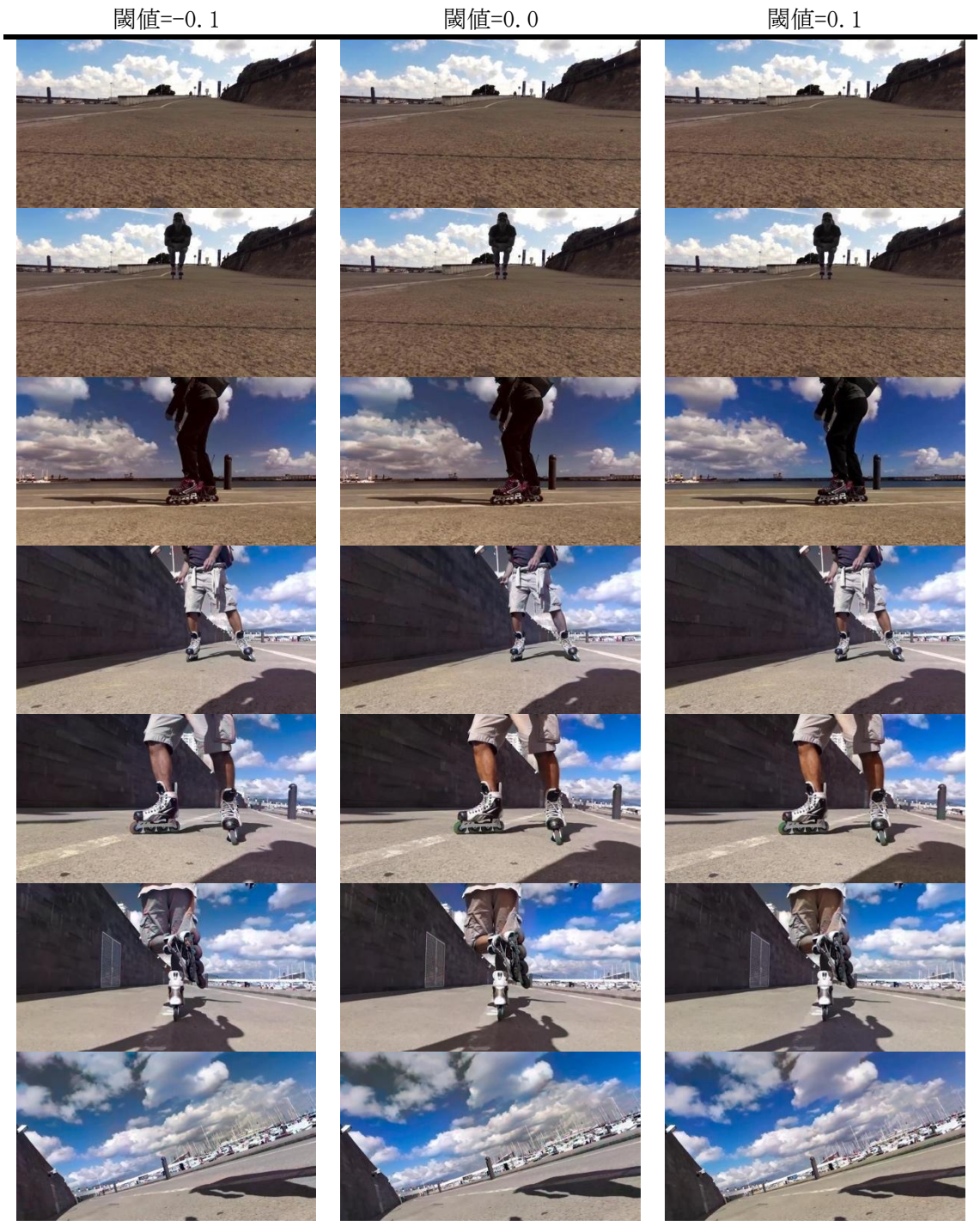


Fig 4-4-5 検証映像④における出力映像フレーム

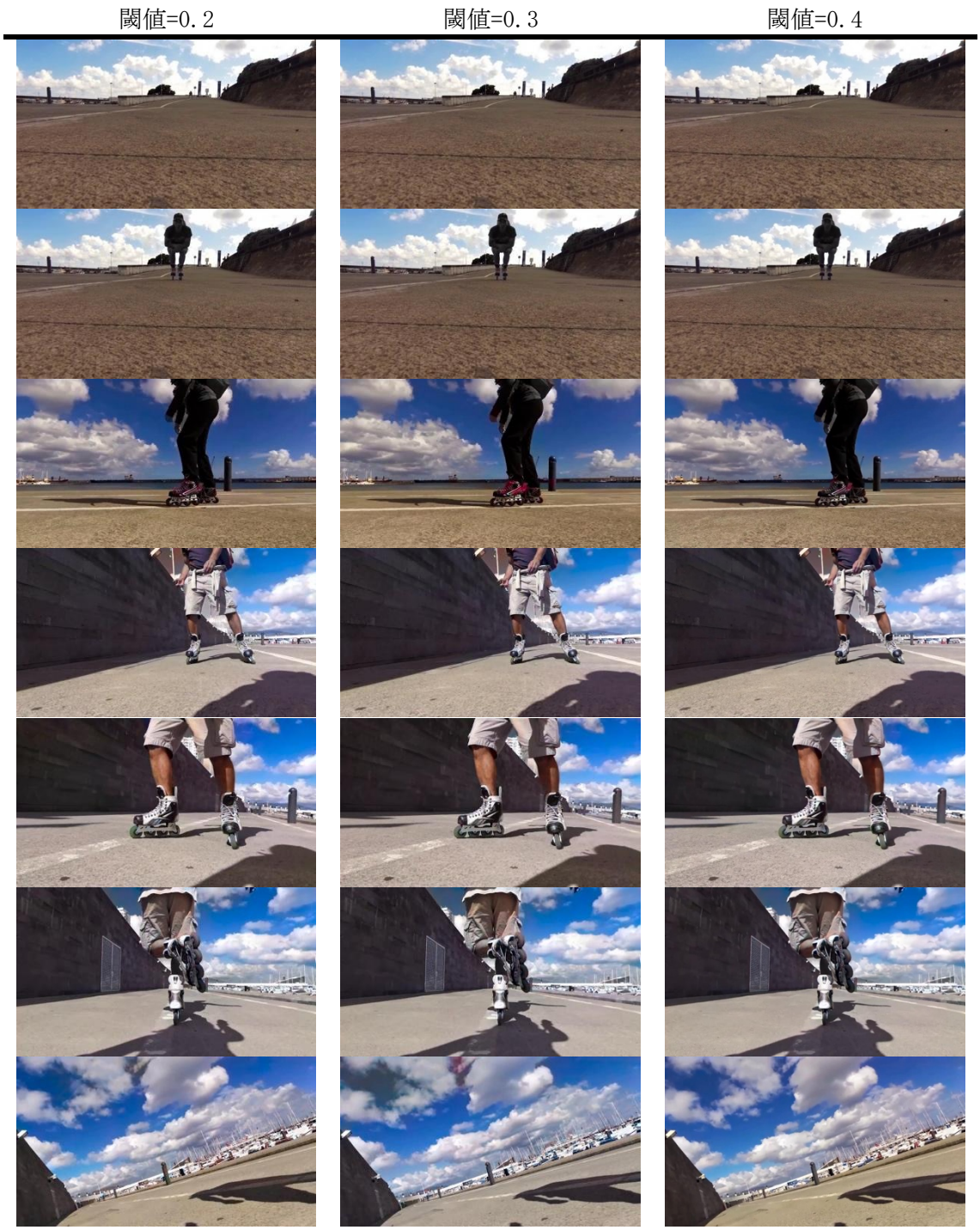


Fig 4.4-6 検証映像④における出力映像フレーム

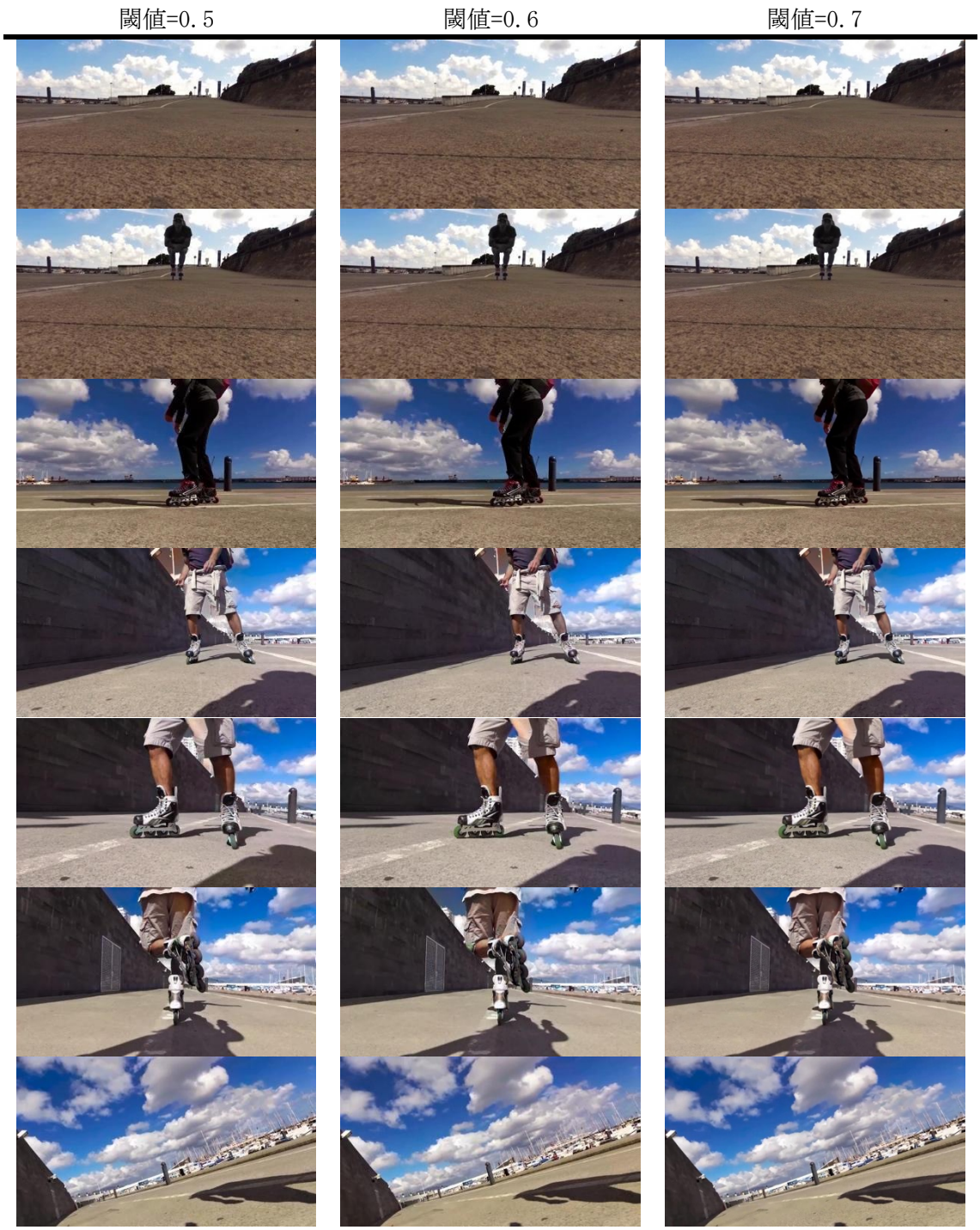


Fig 4.4-7 検証映像④における出力映像フレーム

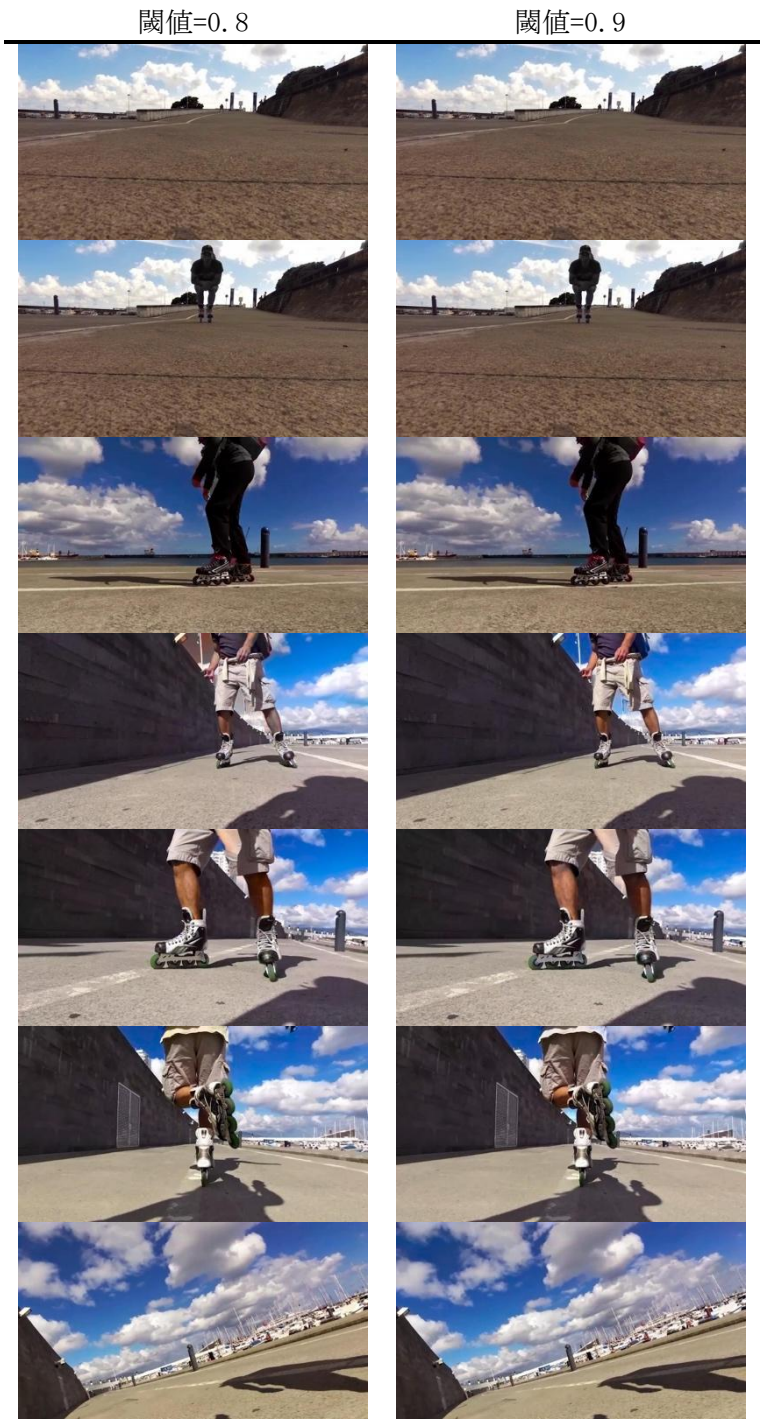


Fig 4.4-8 検証映像④における出力映像フレーム

4.5 検証映像⑤の出力結果

Table 4.5-1: 検証映像⑤における設定閾値とそれに伴う参照画像切り替え数,PSNR,SSIM

閾値	切り替え回数	PSNR［dB］	SSIM
-0.9 から 0.7	0	29.367	0.930
0.8	3	30.717	0.932
0.9	9	30.981	0.931

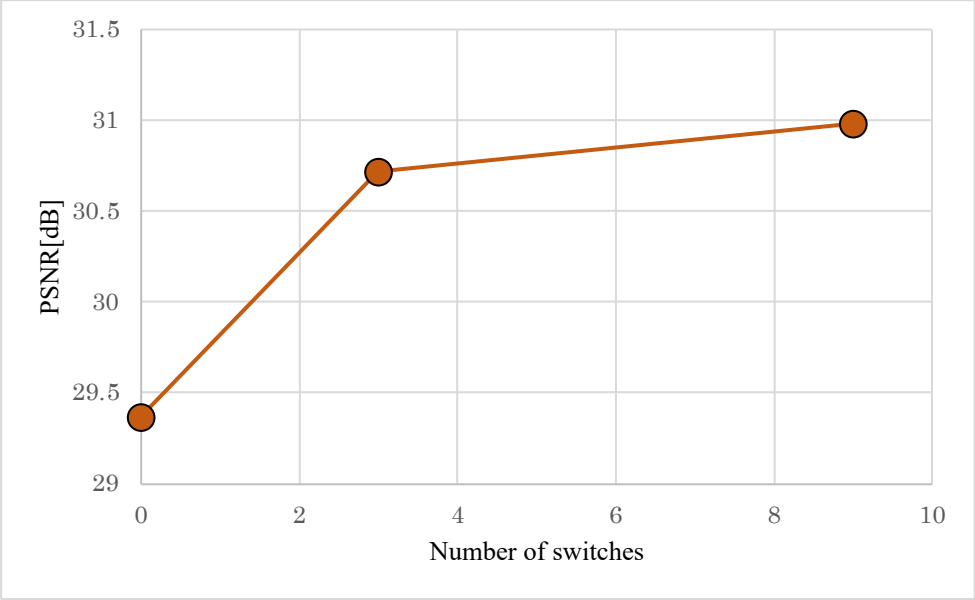


Fig 4.5-1:参照画像の切り替え数と PSNR の関係

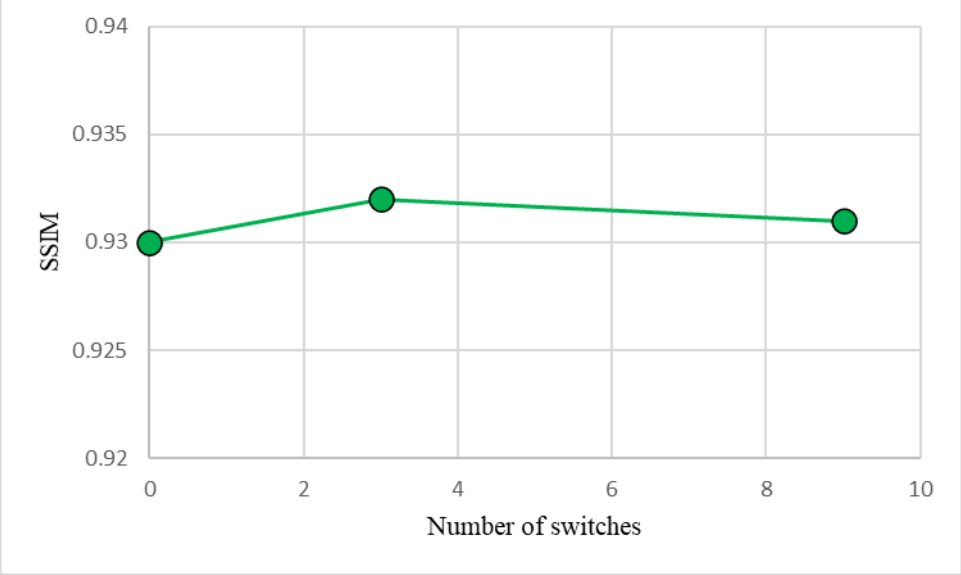


Fig 4.5-2:参照画像の切り替え数と SSIM の関係

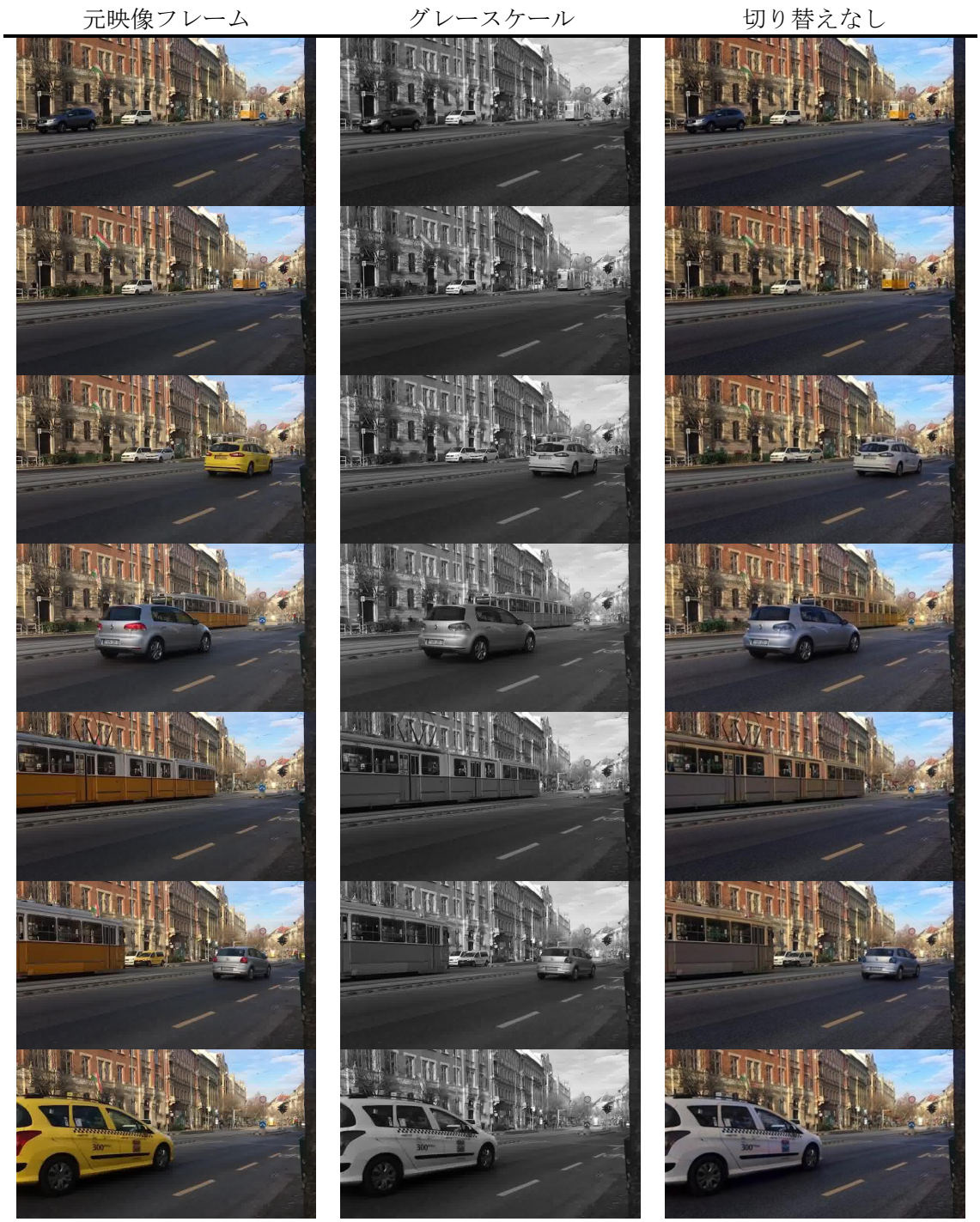


Fig 4.5-3 ：映像データ⑤における出力映像フレーム

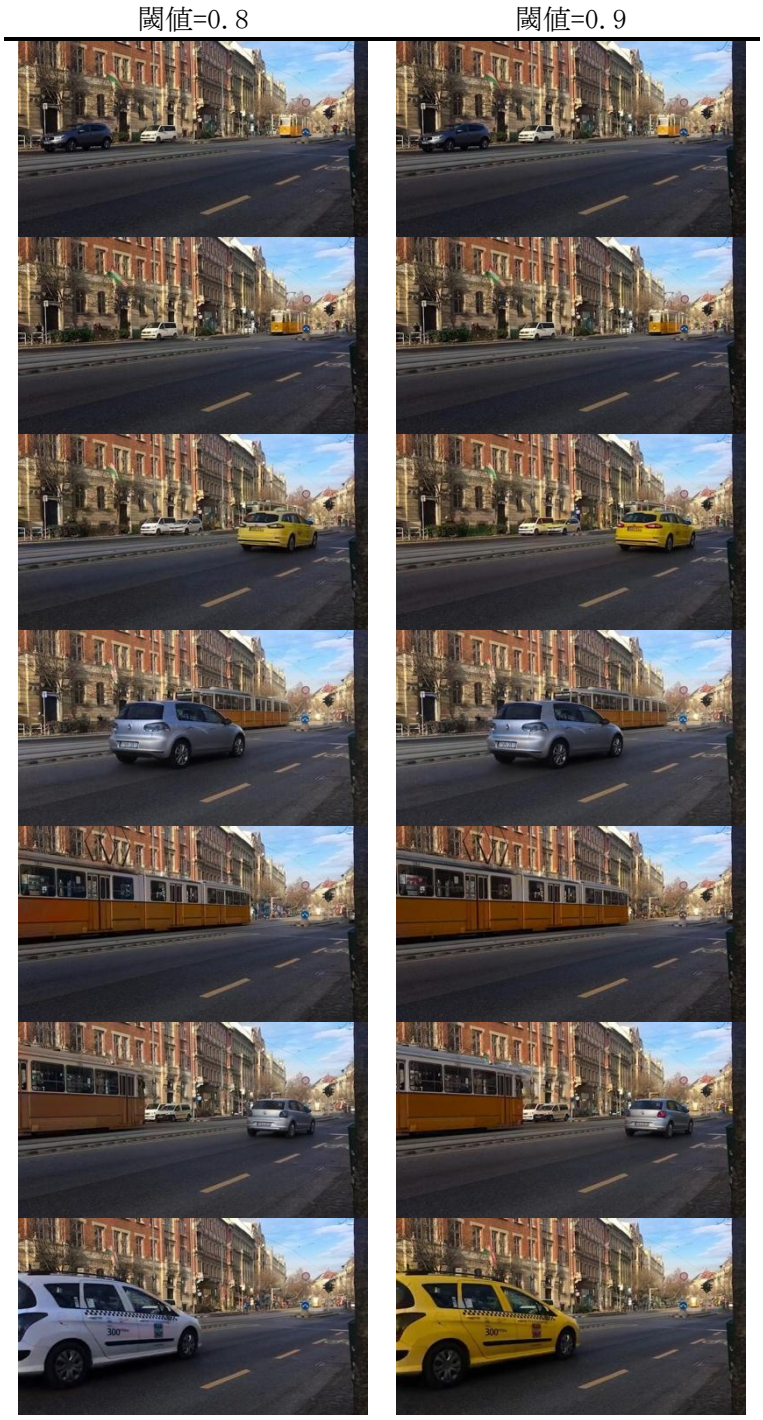


Fig 4.5-4 ：映像データ⑤における出力映像フレーム

4.6 検証映像⑥の出力結果

Table 4.6-1:検証映像⑥における設定閾値とそれに伴う参照画像切り替え数,PSNR,SSIM

閾値	切り替え回数	PSNR［dB］	SSIM
--0.9 から 0.9	0	29.183	0.95

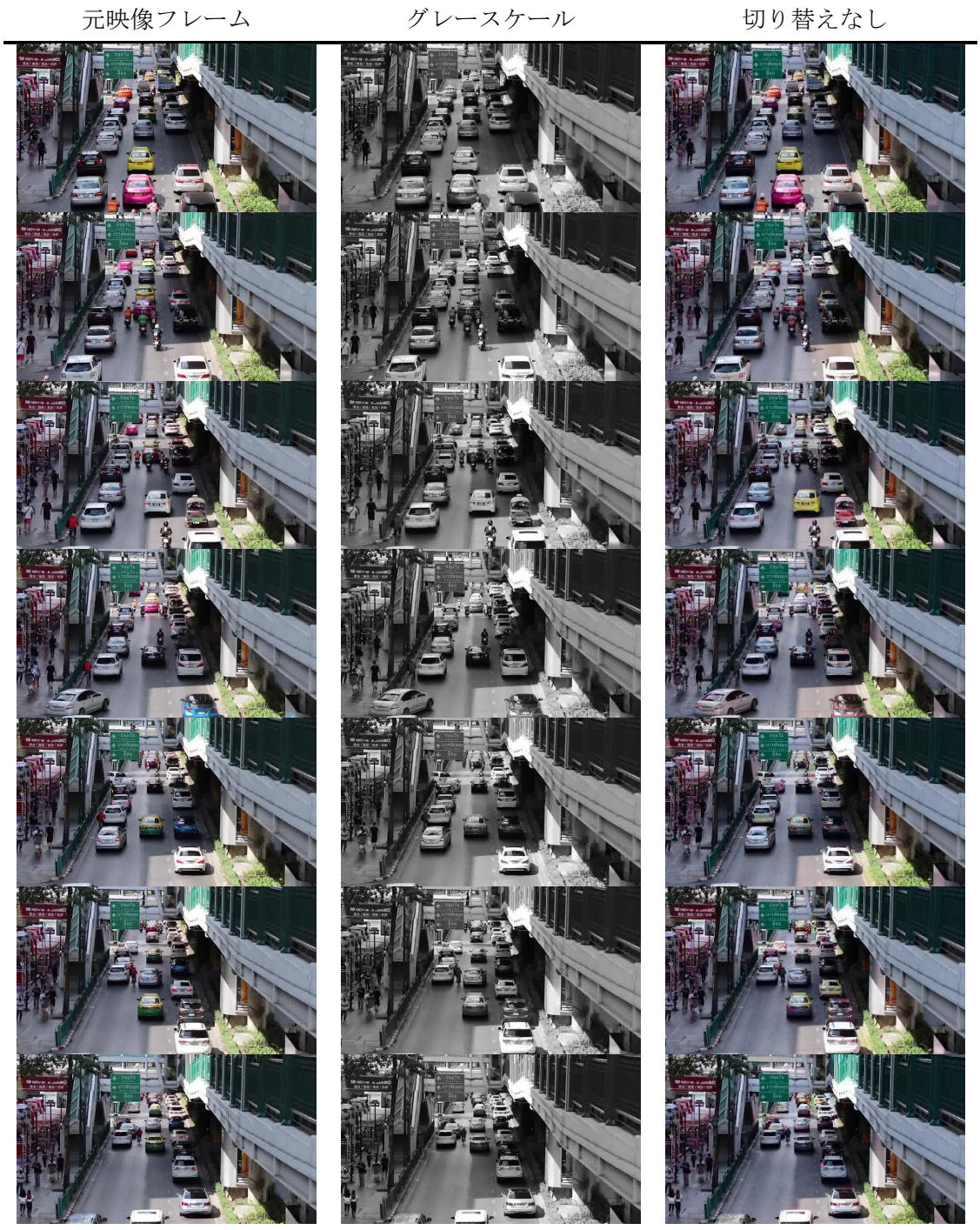
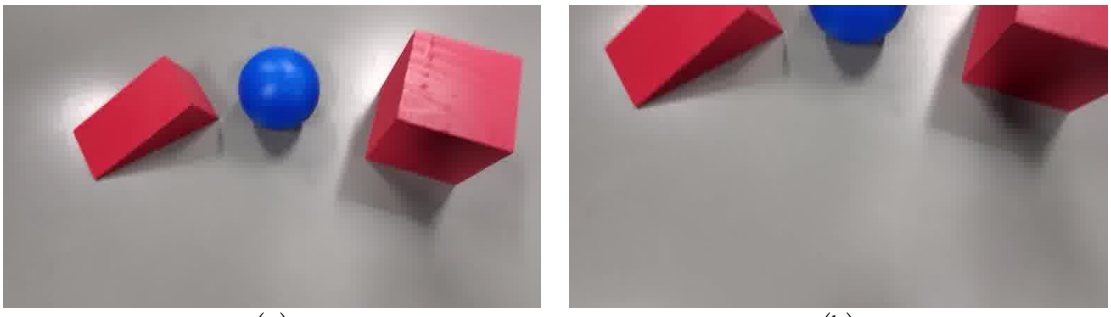


Fig 4.6-1 ：映像データ⑥における出力映像フレーム

5 考察

映像データ①から⑤のカラー化において,参照画像を切り替えないカラー化と比べ,ヒストグラムの相関値による参照画像の自動切り替えが行われたカラー化の方がより高い PSNR 値であり,品質の向上が確認できた.またそれは参照画像の切り替え数が増加するにつれより向上する.

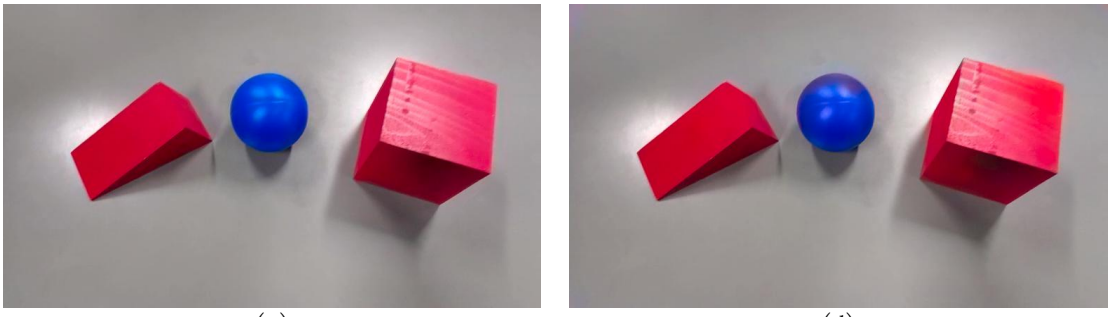
しかし映像データ①の参照画像切り替えのための閾値が 0.6 と 0.7 による参照画像の切り替え数は両者共に同じ 1 であるにも関わらず,PSNR の値は異なっていた. 閾値が 0.6 であるより 0.7 の方が,参照画像の切り替わりが発生しやすい.実際に閾値が 0.6 と 0.7 の参照画像切り替わりタイミングを調査したところ,閾値が 0.6 の場合,切り替えフレームは 39 枚目で,0.7 の場合は切り替えフレームは 23 枚目であった.



(a) (b)

Fig5.1 :映像データ①における閾値 0.6 と 0.7 の参照画像の置き換わりフレーム
(a) 閾値 0.6 において新たな参照画像として置き換わったフレーム
(b) 閾値 0.7 において新たな参照画像として置き換わったフレーム.
以降のカラー化では(a),(b)のフレームがそれぞれ参照画像に用いられた.

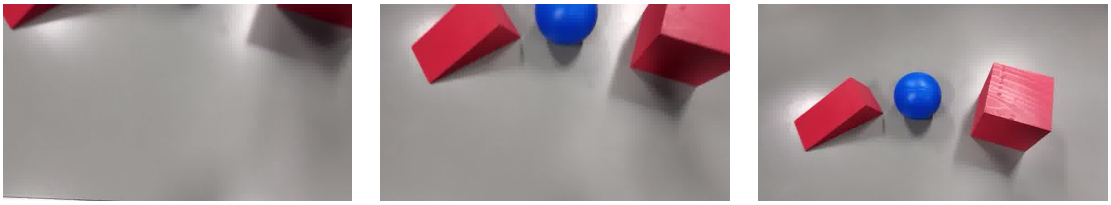
ここで 5-1:(b)の閾値 0.7 で発生した参照画像の置き換えフレームを確認すると,物体の形が完全に確認できる前に切り替えが起きていることがわかる.ここで閾値 0.7 のカラー化フレームを確認すると,新たな参照画像となった 23 枚目のフレームには映っていなかった球体の上半分の色が閾値 0.6 の色と比べ,鮮やかでないことがわかる.



(c) (d)

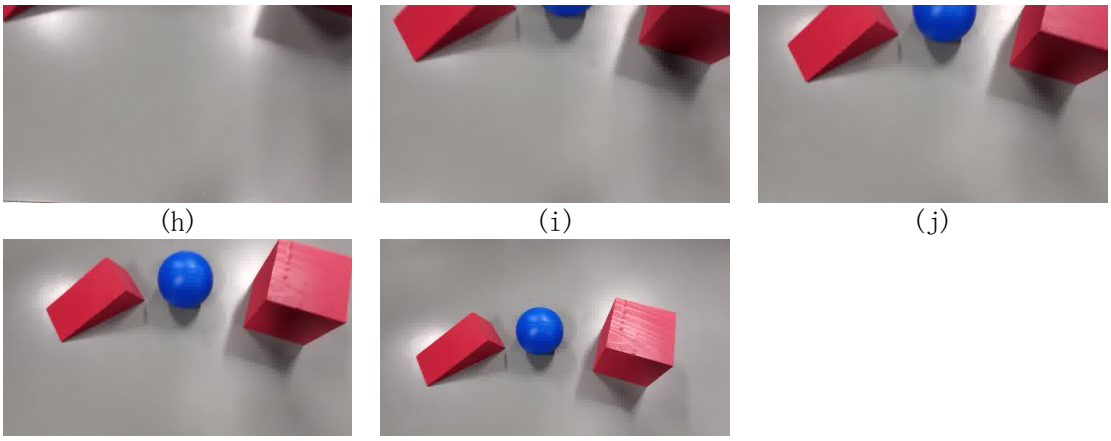
Fig 5.2 :映像データ①における閾値 0.6 と 0.7 のカラー化後フレーム.
(c)閾値 0.6 でのカラー化後フレーム
(d)閾値 0.7 でのカラー化後フレーム

Fig5-2(d)の結果は,Fig5-1(b)の新たな参照画像に球体上半分の情報が存在しなかったためと推測できる.
しかし閾値が 0.8 に変わり,参照画像の切り替えがより多く発生しやすくなったことで,物体の出現により対応できている.閾値 0.9 に関しても同様である.また閾値 0.8 では物体が登場時,青色の球体が赤色にカラー化されていたが 0.9 では本来の赤色にカラー化されていた.
閾値の変更により参照画像の切り替わりが早い段階で発生する.切り替え数が増えるのにつれ,より早いタイミングで正確に映像内物体がカラー化され,品質も向上する.



(e) (f) (g)

Fig 5.3:映像データ①における閾値 0.8 の参照画像の 切り替わりフレーム
(e)フレーム 12 枚目
(f)フレーム 27 枚目
(g)フレーム 65 枚目



(h) (i) (j)
(k) (l)

Fig 5.4:映像データ①における閾値 0.9 の参照画像の切り替わりフレーム
(h)フレーム 9 枚目
(i) フレーム 20 枚目
(j) フレーム 26 枚目
(k) フレーム 35 枚目
(l) フレーム 65 枚目

このような参照画像の切り替えタイミングのずれによる,閾値上昇にも関わらず参照画像の切り替え数の減少であったり,切り替え数増加による PSNR 値減少や不適切なカラー化は映像データ②,③,④でも見られた.そして参照画像の切り替えタイミングのずれによる適切でないカラー化は切り替え数をより多くすることで解決が可能であった.

一方,映像データ⑥では参照画像の切り替わり自体が発生しなかった.

映像データ⑤と映像データ⑥は定点映像である.映像データ⑤では参照画像切り替えが発生したが,映像データ⑥では参照画像の切り替わりが発生せず,車等の複数物体は元映像フレームの色と異なるカラー化がされていた。

(m) (n)

Fig 5.5 :映像データ⑥における出力結果.

(m) 元映像のフレーム

(n) 参照画像切り替えなしカラー化後フレーム

映像データ⑥において最初に参照画像であった第 1 フレームと映像フレーム中に大きな色の違いはないとされ,参照画像の切り替えが発生せず全てのフレームにおいて第 1 フレームが参照画像として使用された.しかし参照画像に存在しない物体がフレーム中に現れたためカラー化により元の色の復元ができなかったと考えられる.

また確認のため,映像フレームごとのヒストグラムの相関値を算出した.

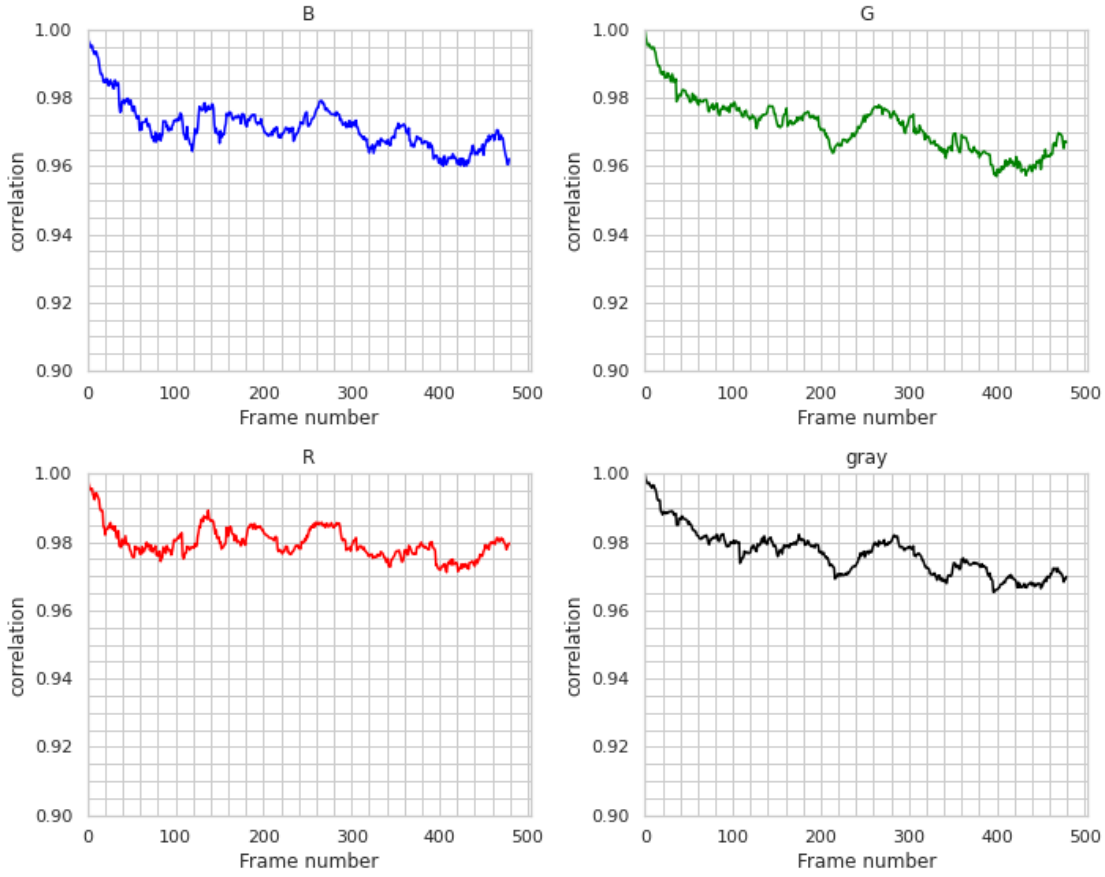


Fig 5.6:映像データ⑥のフレームごとのヒストグラム相関値,
参考画像を切り替える際には グレースケールは使用していないが参考のため算出.

Fig5.6 より,全てのフレームでヒストグラムの相関値の動きは 1.00~0.96 の範囲内であり,大きな動きは存在していないことがわかる.実際に映像中では,参照画像には存在しない色の車が出現していたものの,フレーム全体としては大きな色の変化とは言えないものだった.

ヒストグラムの相関値による切り替えは,フレーム画像全体としての色の変化には対応できたが,小さな色の変化には対応できないことが考えられる.

SSIM の値は,切り替えが起きた映像②から⑤において PSNR と同じく上昇傾向であった.しかし,映像①における SSIM は,切り替えが起こらなかった時点が一番高い値であった.これは PSNR の傾向, カラー化後出力フレームの結果と反するものであり,必ずしも評価指標は人間の視覚に対応するものではないことが判明した.より人間の知覚を反映した評価を行う必要がある.

6 結論

本論文では, 白黒映像から元映像の色を復元するために, 参照画像に基づいたカラー化を行うディープラーニングを使用し, さらにヒストグラムの相関値による参照画像の自動切り替えを行うことで元映像の色の復元を試みた. 本研究の主な結論を以下にまとめる.

・検証映像データ 6 つ中 5 つが, 参照画像の切り替わりがない場合の PSNR より, ヒストグラムによる参照画像切り替えを行った場合の方が PSNR は高い値となった. また閾値を変更した際には, 閾値の値が上昇するにつれて, 参照画像の切り替え回数が増加し, PSNR も上昇した. また品質も上昇したといえる.

・ヒストグラムによる参照画像の切り替えは, フレーム画像全体のヒストグラムの相関値を用いて行われるため, フレーム画像内の僅かな色の変化には反応しない.

より小さな物体の動きや色の変化に対応するには, 画像全体のヒストグラムを利用するのではなく, 細かい領域に分割し, それぞれの領域ごとにヒストグラムを適用するなどが考えられる.
また人間の知覚により対応した評価を行う必要がある.

7 参考文献

[1] Fatima,A.,Hussain,W.& Rasool,S.” Grey is the new RGB: How good is GAN-based. image colorization for image compression?” Multimed Tools Appl 80,3775-3791

[2] Z.Pan,F.Yuan,J.Lei,S.Kwong,Video compression coding via colorization: a generative.adversarial network (gan)-based approach, arXiv preprint arXiv:1912.10653,2019.

[3] S.Iizuka,E.Simo-Serra,and H.Ishikawa,” Let there be color! joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification,” ACM Trans. Graph.,vol.35,no. 4, pp. 1-11, 2016.

[4] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros, “Real-time user-guided image colorization with learned deep priors,” arXiv preprint arXiv:1705.02999,2017.

[5] B.Zhang, M. He, J. Liao, P. V. Sander, L. Yuan, A. Bermak,and D.Chen, “Deep exemplar-based video colorization,” in Proceedings of the IEEE. Conference on Computer Vision and Pattern Recognition, 2019, pp. 8052-8061.

[6] 山下隆義 (2018) 『イラストで学ぶ ディープラーニング 改訂第2版』講談社

[7] ラシュカ,セバスチャン・ミルジャリリ,ヴァヒド(2020)『[第3版]Python機械学習プログラミング達人データサイエンティストによる理論と実践』 クイープ訳,福島真太郎監訳,インプレス

[8] Simonyan,K.,&Zisserman,A.(2014).” Very deep convolutional networks for large. scale image recognition.” arXiv preprint arXiv:1409.1556

[9] Levin,Anat & Lischinski,Dani & Weiss,Yair.(2004).” Colorization using. Optimization.” ACM Transactions on Graphics. 23. 10.1145/1015706.1015780.

[10] Welsh,Tomihisa & Ashikhmin, Michael & Mueller,Klaus.(2002). “Transferring.Color to.Greyscale Images.” ACM Trans.Graph..21.277- 280.10.1145/566570.566576.

[11] D.Chen, J. Liao, L. Yuan, N. Yu, and G. Hua, “Coherent online video style transfer,” in Proceedings of the IEEE International Conference on Computer Vision, pp. 1105-1114, 2017.

[12] ディジタル画像処理編集委員会著(2020)『ディジタル画像処理[改訂第二版]』画像情報教育振興協会.

[13] Nisreen.I.Radwan,Nancy.M.Salem,and Mohamed.I.El Adawy,(2012)“Histogram Correlation. for Video Scene Change Detection,”in Advances in Computer Science,Engineering&Applications,vol.166,D.C.Wyld,J.Zizka,andD.Nagamalai, Eds.,pp.765-773, Springer Berlin Heidelberg, Berlin, Heidelberg,

[14] 大町真一郎・陳謙・大町方子・宮田高道・長谷川為春・早川吉彦・加瀬澤正・塩入諭(2014)『未来へつなぐデジタルシリーズ 28 画像処理』白鳥則郎監修,共立出版.

[15] pixabay.com,< https://pixabay.com/ja/videos/>2022 年 1 月 20 日アクセス

[16]“スケート-キャスター-スケーター-3674”,<https://pixabay.com/videos/id-3674/ >2022 年 1 月 20 日アクセス

[17]“トラム-ブダペスト-ハンガリー-31013”,https://pixabay.com/videos/id31013/> 2022 年 1 月 20 日アクセス

[18]“タイ-車-道-街-輸送-30001”,<https://pixabay.com/videos/id-30001/> 2022 年 1 月 20 日アクセス

8 謝辞

本論文は多くの方々のご協力により完成させることができました.
本研究を実施するにあたりミケレット・ルジェロ教授には研究や発表についての熱心なご指導を賜りました. 深く御礼申し上げます.
また, 深層学習や画像処理に関して, 多くの知識やご助言いただいた孫哲研究員をはじめとする理化学研究所画像情報処理研究チームの方々に厚く御礼申し上げます.
そして日々の議論を通じて多くの知識やアドバイスを頂戴いたしましたミケレット研究室の皆様には深く感謝いたします.
最後に家族や友人, 本論文執筆に携わってくれた全ての方々に感謝の意を表し, 謝辞といたします.