6ZD-02

動画中の特定人物に注目した対話的ダイジェスト制作ツール

山下 紗季 † 伊藤 貴之 † Tobias Czauderna‡ Michael Wybrow‡
† お茶の水女子大学 ‡Monash University

1. 概要

本報告では，映像内に登場する人物に注目したダイジェスト動画生成を支援する一手法を提案する．本研究におけるダイジェスト動画の定義は，与えられた動画群からユーザが指定した人物が映るシーンを検出して連結させたものである．

本手法では入力映像を多数のショットに分割し，その各々に顔画像認識を適用する．その結果として，特定人物が確実に含まれると判定されたショットを「正解ショット」とし，ダイジェスト映像を構成するショットとする．さらに，正解ショットのいずれかに対して一定以上の類似度を有するショットを「候補ショット」とする．候補ショットは指定された人物を含む可能性はあるが確定的ではないショットと考えることができる．本手法ではユーザインタフェースを介して候補ショットをユーザに提示し，ダイジェスト映像に含まれるべきショットを選択させる．動画像処理によって自動選択された正解ショットとユーザによって選択された候補ショットを組み合わせることで，少ない操作で満足度の高いダイジェスト動画を生成する．

2. 関連研究

ビデオから特定人物を検出する手法として，まず Tapaswi らの手法 [1] があげられる．この手法はドラマ映像を題材として，顔識別結果に加えて衣服と話者識別の情報を組み合わせて登場人物のラベル付けを行う．

動画編集を支援するためにユーザインタフェースを生成する手法として，土田らの手法 [2] をあげる．このシステムでは，複数のカメラから同時に撮影されたダンス映像を自動編集するだけでなく，ユーザインタフェースを生成することで好みのダンス映像に調整できる仕組みを提供する．

3. ユーザインタフェースの設計

本報告では図 1 のようなユーザインタフェースを提案する．ダイジェスト動画へ採用することが決定されたショット（以下「採用済みショット」と呼ぶ）を画面の 4 辺に連結して配置し，その内側に採用候補となるショットを配置する．ユーザは採用したいショットを採用済みショット列の任意の位置へドラッグすることにより，ショットの採用と挿入ができる．候補となるショットから採用済みショットへ接続されている線分は，システムが推薦する挿入位置を表す．

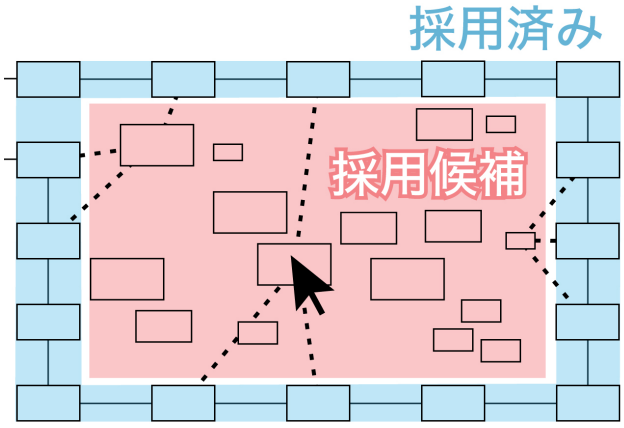

図 1: ユーザインタフェースの設計

4. ユーザインタフェース生成の前処理

本章では，指定人物の自動的な検出やユーザインタフェース生成のための情報を取得する処理について述べる．

4.1 ショット分割

まず，入力された動画をショットに分割する．ショットとは，場面が大きく変化するカット点に挟まれた連続したフレームを指す．カット点を検出し，検出結果にもとづいて入力動画からショットを切り出す．

4.2 顔検出と顔識別にもとづく得点付与

続いて各ショット中の顔領域から指定人物を含む可能性を推定し，ショットに得点を与える．

はじめにショット中の顔領域を検出する．顔領域検出には Microsoft Azure[3] の Media Services を用いた．顔検出されたショットについては，ユーザに指定人物の顔画像を入力させ検出された顔領域との類似度を範囲 $[0.0, 1.0]$ の実数で算出し，それを得点とする．類似度は Microsoft Azure の Face API を用いて算出する．顔検出されなかったショットについては，顔検出されたショットの得点をもとに得点を算出する．得点を求めたいショット A の得点を P_A として，ショット A と顔検出できたショット群 B_i との類似度をそれぞれ求める．類似度を $Sim(A, B_i)$ としたときに，以下の式で表される実数をショット A の得点を P_A とする．

$$P_A = max(P_{B_i} Sim(A, B_i)) \quad (1)$$

これにより顔領域の条件の差を吸収したショット選出を可能にする．

そして $[0, 1]$ の間に閾値を 2 つ定め，それらを s, t $(s < t)$ としたとき，$P_A < s$ となるショットは指定人物が存在しないであろうとしてダイジェスト動画に組み込むショットの候補から除外する．ここで，$P_A > t$ となるショットは確実に指定人物を含んでいるとして，正解ショットとする．

An Interactive Digest Movie Creation Tool Focusing on Specific Persons
† Saki Yamashita, Takayuki Itoh
‡ Tobias Czauderna, Michael Wybrow
Ochanomizu University (†)
Monash University (‡)

一方，$s \leq P_A \leq t$ であるショットは指定人物を含む可能性はあるが確定的ではないとし，候補ショットとする．

4.3 候補ショットの挿入位置の推薦

続いて候補ショットの各々について，どの正解ショットの前後に挿入するのがふさわしいかを判定する．ここで，ある候補ショットを C，正解ショット群を D_i とし，挿入の推薦度を $R(C, D_i)$ とする．CD_i 間の画像的類似度を Im，入力動画における時間的近接度を Tm としたとき，以下の式によって $R(C, D_i)$ を求める．

$$R(C, D_i) = Tm + (1 - Tm)Im \tag{2}$$

$$Tm = (Dist(C, D_i) - 1)^a \tag{3}$$

$Dist(C, D_i)$ は C と D_i のショット番号の差を求め，その絶対値を全体のショット数で正規化したものである．$Dist$ が 0 に近づくと時間的近接度 Tm が大きくなる．現時点での我々の実装では $a = 6.0$ としている．これによって画像内容が類似している正解ショットの前後のみならず，時系列的に近接している正解ショットの前後も挿入位置として推薦することができる．

5. ユーザインタフェースの生成

本章では，ショット連結順を確認しながらショット選択ができるユーザインタフェースを提案する．ユーザインタフェースは，各ショットのサムネイルをノードとしたグラフとして生成される．グラフのエッジは，時間的に隣接する 2 つの採用済みショットの連結，および候補ショットに対してシステムが推薦する挿入位置への連結を表現する．

5.1 採用済みショットの配置

4.2 節で説明したとおり，正解ショットはあらかじめダイジェスト動画に採用される．ユーザインタフェースの生成に際してまず，この正解ショットのみを対象として仮の再生順を決定し連結する．そして図 1 のように，正解ショットを画面の 4 辺に連結して表示する．

5.2 候補ショットの配置

続いて仮連結された正解ショットの内側に候補ショットを配置する．候補ショットの画面上の位置の算出には力学指向ノード配置手法を用いる．候補ショットとシステムが推薦する挿入位置はエッジで接続されているため，ユーザは各候補ショットの位置から挿入位置を推察することができる．サムネイルの大きさは 4.2 節で求めた得点に比例した大きさとする．

5.3 ショットの挿入と削除

ユーザは候補ショットを正解ショット列の任意の位置にドラッグすることでショットを挿入する．ショットの挿入が行われると，ユーザインタフェースは採用済みのショット列を更新し 5.1 節と同様の処理によって再配置する．また，候補ショットをダブルクリックするとそのショットを削除する．

6. ユーザインタフェースの実行例

本章ではユーザインタフェースの実行例を紹介する．この例では入力動画として 1 本のミュージックビデオを使用し，$s = 0.05$，$t = 0.8$ とした．正解ショットの連結順序は入力動画における時系列順とした．ユーザインタフェースの実装には cola.js[4] を適用した．

プログラムを実行し，マウスボタンを候補ショットのサムネイル上で押下すると，図 2 のように挿入位置の候補となるショットへのエッジが表示される．

図 2: ユーザインタフェースの実行例

7. まとめと今後の課題

本報告では，特定人物に注目したダイジェスト動画生成を支援する手法の一環として，自動判別とユーザによる選択を組み合わせたショットを選出するためのユーザインタフェースを提案した．

今後の課題としては，候補ショットをクラスタリングして表示するなどユーザインタフェースの拡充があげられる．また得点を改善するための手法として，一般物体認識を用いてショットに人物が含まれるかどうかを判定し，その結果を反映することがあげられる．

参考文献

[1] Makarand Tapaswi, Martin Bäuml and Rainer Stiefelhagen, “Knock! Knock! Who is it?” probabilistic person identification in TV-series, Computer Vision and Pattern Recognition (CVPR), pp. 2658–2665, 2012.

[2] 土田修平, 深山覚, 後藤真孝, 多視点ダンス映像のインタラクティブ編集システム, 第 25 回インタラクティブシステムとソフトウェアに関するワークショップ (WISS 2017) pp. 41–46, 2017.

[3] Microsoft: Microsoft Azure Cloud Computing Platform & Services （確認日 2019/1/4） https://azure.microsoft.com/ja-jp/

[4] Tim Dwyer: cola.js: Constraint-based Layout in the Browser （確認日 2019/1/4） https://ialab.it.monash.edu/webcola/