4G1-3

手話翻訳のためのモーションベクトル化による パターン識別アルゴリズム

The pattern recognition algorithm by motion vectorization for sign language translation

藤本 美加[*1]
Mika Fujimoto

井上 聡[*1*2]
Satoru Inoue

[*1] 埼玉工業大学大学院
Graduate School of Engineering, Saitama Institute of Technology

[*2] 埼玉工業大学
Saitama Institute of Technology

Obtains the sign language operation from an external optical device, it is intended to translate the natural language by the vectorization. We propose the algorithm of the pattern recognition based on the motion vectorization by using the longest common subsequence.Furthermore language model is adopted to our proposed algorithm to achieve the higher recognition accuracy.

1. はじめに

1.1 研究背景

目が見える，耳が聞こえるということは当たり前のように思えるが，先天的，後天的に身体に障害を持つ人も多くいる．そして点字や手話，車いすなど，障害を持つ人々の為に開発されたものが世の中には多数存在する．しかし，それらは生まれた時から自由に扱えるものではない．特に障害を持たずに生活が可能な健常者は，私生活の中でもそれらを必要とする機会は非常に少ない．もし健常者と障害者との間でコミュニケーションを必要とした場合，それらの知識がないということはコミュニケーションを取る上で大きな障害になる．また，急に障害を持ってしまった場合，それらの知識を一人で身に付けることは時間がかかる．上記のような問題を解決するシステムは，視覚障害者でも健常者でも同じ紙で文字が読めるように開発された点字プリンタなどが開発されている．聴覚障害者のためのコミュニケーション支援システムも開発されているが，ストレスを感じやすいものが多い．また，支援システムと近いものとしてジェスチャ識別が挙げられるが，ジェスチャ識別には赤外線センサや距離画像センサを用いたものが多い．それらは身近なものではなく，入手するには困難である．家電量販店に並んでいるものでも，価格は万単位となってしまうため決して安価なものではない．そのため，手話の学習支援や生活に役立てるためにはあまり向いていない．

本研究では上記のような問題を解決する為に，赤外線カメラなどよりも比較的安価な単眼カメラを用い，入力された手話の映像から手話の意味の翻訳を行う手法を提案する．

1.2 先行研究

先行研究では手話翻訳には指先の位置を特定することが必要だと考え，指先位置を特定するシステムの研究が行われていた．手の向きごとに識別機を分け，指先の学習データをもとに指先と思われる位置を抽出，手の重心から指先の候補の位置を測り，手の重心から遠い指先候補 5 つを指先として判定していた．この手法で実験を行った結果を図 1.2.1 に示す．指先位置を判定することができた．しかし指先と手首の距離の差があまりない場合，手首部分が指先として検出されてしまう，また，指の付け根が指先として検出されるなどの誤認識が発生した．

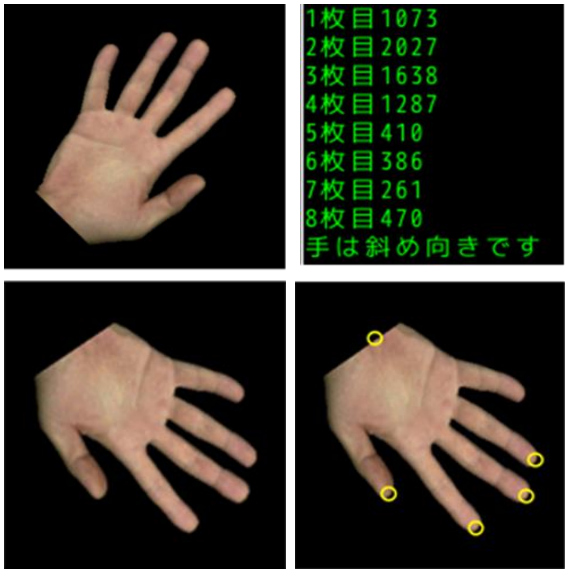

図 1.2.1 手の傾きと指先位置の検出結果

上記の問題を解決するため，手の輪郭に一定間隔で点を打ち，3 点の角度が閾値内であった場合は指先とした[藤本 14]．その結果，どの角度からも指先の検出はできたが，切り取られた画像の手首の部分に誤認識が起きる問題が発生した．

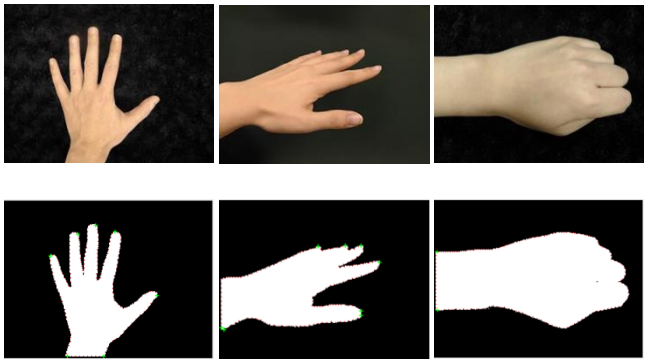

図 1.2.2 指先位置の検出結果

連絡先：藤本 美加，埼玉工業大学大学院，埼玉県深谷市普済寺 1690 番地，E-mail：popcon0910@gmail.com

1.3 本研究の目的

手話の動作を識別するには指先の細かい動きを判別する事も重要であるが，身振り手振りなどのジェスチャが主である．そのため，手話を翻訳するには，指先位置の特定だけでなくジェスチャ認識が必要だと考えた．そこで本研究では，体の動きをベクトル化し，パターン分類をすることで手話の翻訳を行うことを目的とする．

2. 提案手法

本研究で提案する手法の処理手順を図 2.1 に示す．

今回はシステム構築ではなく提案アルゴリズムの検証のため，画像処理部分のアルゴリズムの検証は手操作で行った．

2.1 動作の分割とベクトル化

手話の動作を識別，翻訳するにあたり，本研究では動作のベクトル化を行う手法を提案している．動作のベクトル化を行うに

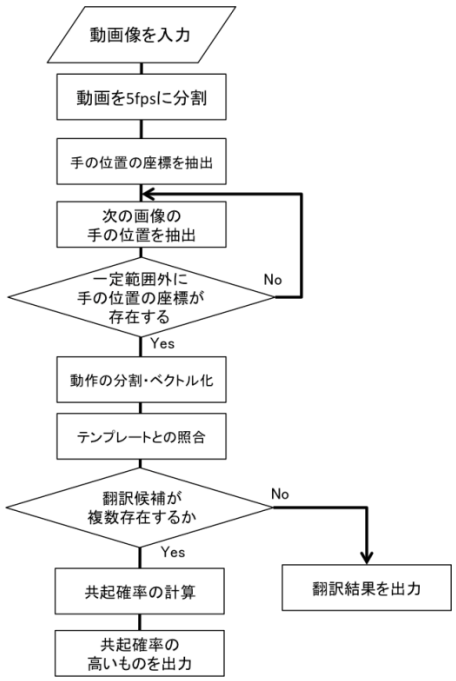

図 2.1.1 本研究での提案手法の処理手順

あたり，取得した全体の手話動作を細かく分割する必要がある．そのため，今回は動画を 5fps の間隔で分割し画像として保存，手の位置の座標を連続して取得する．閾値として角度 θ を設け，その閾値外に手の座標の位置が検出されるまでの動作を一動作とする．閾値外に手の座標が検出された場合，その直前までの動作を一動作とし，閾値外に検出された手の座標から，再度動作の取得を開始する．上記の方法を用いることにより，動作の速度が人により異なっていた場合でも，同等に動作を分割することが可能となる．

分割された動作が，右手と左手の「どちらが動いていたか」「手がどのように動いたか」「手の動作の始点の位置」「反対の手の動作状況」の 4 項目を，それぞれ英数字を用いてベクトル表現する．表現方法は，左右どちらの手が動いていたかを右手を「RIGHT」左手を「LEFT」，手がどのように動いたかは 8 方向にそれぞれ数値を図 2.2 のように振り分け，手の始点の位置は顔の前から手の動作が始まった場合は「FF」，顔の横からは「FW」，胸の前からは「C」，腹の前からは「B」とし，反対の手の動作状況は，動いている場合を「MV」，止まっている場合を「ST」とした．

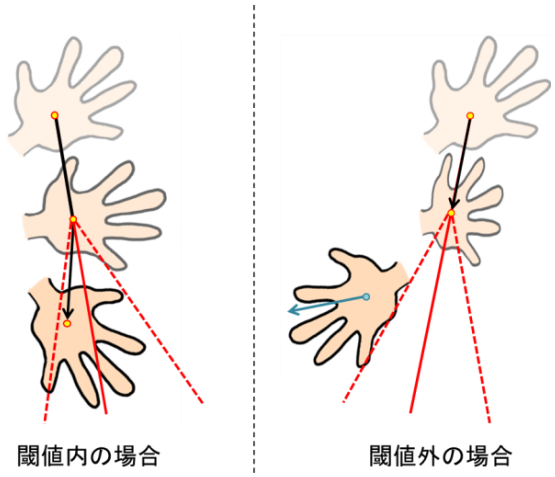

図 2.1.2 動作の分割イメージ

例えば，右手が始点よりも右上方向に動き，且つ右手の動作が顔の横から始まり，左手も動いていた場合は表 2.1 のような表現となる．

分割された動作が，右手と左手の「どちらが動いていたか」「手がどのように動いたか」「手の動作の始点の位置」「反対の手の動作状況」の 4 項目を，それぞれ英数字を用いてベクトル表現する．表現方法は，左右どちらの手が動いていたかを右手を「RIGHT」左手を「LEFT」，手がどのように動いたかは 8 方向にそれぞれ数値を図 2.2 のように振り分け，手の始点の位置は顔の前から手の動作が始まった場合は「FF」，顔の横からは「FW」，胸の前からは「C」，腹の前からは「B」とし，反対の手の動作状況は，動いている場合を「MV」，止まっている場合を「ST」とした．例えば，右手が始点よりも右上方向に動き，且つ右手の動作が顔の横から始まり，左手も動いていた場合は表 2.1 のような表現となる．

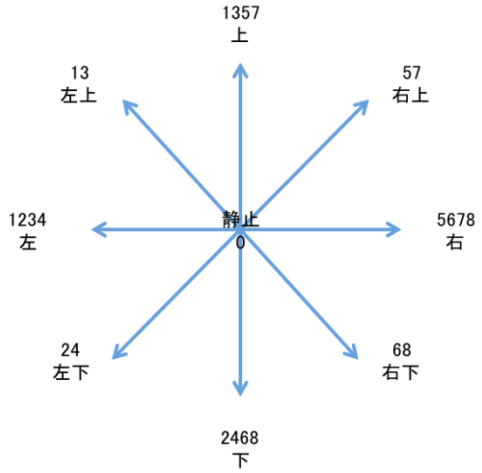

図 2.1.2 8 方向のベクトル表現

表 2.1 動作のベクトル表現例

どちらの手か	手がどのように動いたか	手の始点の位置	反対の手の状況
RIGHT	57	FW	MV

上記の表現方法で手話動作をベクトル表現し，テンプレートの作成を行った．手話の表現は，単語同士の掛け合わせで一つの言葉になることが多い．例えば，「おはようございます」は，「朝」＋「あいさつ」で成り立ち，「初めまして」は，「初めて」＋「会う」で成り立つ．また，一単語は基本的に短い動作で表される．そのため，動作を最小単位の単語で分割し，単語ごとにベクトル表現したものをテンプレートとした．

2.2 テンプレートとの照合

前項でベクトル化された動作を，最長共通部分列(LCS: Longest Common Subsequence)を用いて照合していく．

最長共通部分列とは，2 つの文字列において最も長い文字部分列のことを指す．LCS が長いほど，その 2 つの文字列は類似していることとなる．部分列は文字列とは異なり，前の部分と連続している必要はない．LCS は次式で導き出すことができる．

$$L(a_i, b_j) = \begin{cases} 0 & if(a = 0\ or\ b = 0) \\ \max\left(L(a_{i-1}, b_j), L(a_i - b_{j-1})\right) & if(a_i \neq b_i) \\ L(a_{i-1}, b_{j-1}) + 1 & if(a_i = b_i) \end{cases}$$

a=ARTIFICIAL と b=ARTISTIC の 2 つの文字列を例に挙げる．この 2 つの文字列の最長共通部分列は「ARTIIC」で長さは 6 となる．これを用いることにより，ベクトルの長さが異なっていても類似度を調査することが可能である．

また，最長共通部分列は下記のように 2 次元配列と動的計画法を用いると簡単に導き出すことが可能である．

最初に文字列 a, b それぞれの文字数+1 の行列を用意する．一方の文字列を第一行，一方の文字列を第一列に挿入し，左から右，上から下へと数字を入力していく．行列の文字が一致した場合は左，上，左上の数字の中から最も大きい値に 1 を足したものを入力し，そうでない場合は左，上，左上の中から最も大きい数字をそのまま入力する．そうすると表 2.2.1 のような表が出来上がる．それぞれの値の最も右上の部分の値の文字を取り，末尾から並べていくと最長共通部分列が出来上がる．

表 2.1.1 表を用いた最長共通部分列の算出

		A	R	T	I	S	T	I	C
	0	0	0	0	0	0	0	0	0
A	0	**1**	1	1	1	1	1	1	1
R	0	1	**2**	2	2	2	2	2	2
T	0	1	2	**3**	3	3	3	3	3
I	0	1	2	3	**4**	4	4	4	4
F	0	1	2	3	4	4	4	4	4
I	0	1	2	3	4	4	4	**5**	5
C	0	1	2	3	4	4	4	5	**6**
I	0	1	2	3	4	4	4	5	6
A	0	1	2	3	4	4	4	5	6

2.2 共起確率の計算

動画像のみのパターンの判別では，人により動作に差が出てしまい，翻訳結果が複数になってしまう恐れがある．そのため，本研究では動画像での複数の翻訳結果が出てしまった場合には単語 N-gram を用いて翻訳結果を出力する．

本研究では名古屋大学が公開している名大会話コーパスをもとに，工藤拓氏が開発したオープンソースの形態素解析エンジン「MeCab」を利用し名詞，動詞，形容詞の原形のみを抜き出したものを N-gram のコーパスとした．

・N-gram

クロード・エルウッド・シャノン(Claude Elwood Shannon)が考え出した言語モデルであり，自然言語処理では頻繁に利用される．ある文字列の中で，N 個の文字列がどの程度出現するか，文章を N 文字，または N 単語ずつに分割し調査する．N には任意の数字が入り，一般的に 2-gram(bi-gram)，3-gram(tri-gram)が用いられることが多い．

例文として，「今日はいい天気ですね」を用いて 3-gram で分割すると，「今日は」「日はい」「はいい」「いい天」「い天気」「天気で」「気です」「ですね」「すね」「ね」と分けられる．この分けられた文字列の中で隣接した文字が，共起頻度が高いと言える．本研究では，これを一文字ずつではなく一単語ずつに分割し，N-gram を用いて共起確率を算出した．

3. 結果

3.1 結果

上記の手法で実験を行った結果を以下に示す．

まず，手話動作の分割とベクトル化，テンプレートとの照合の実験においては，手話の教本を参考にテンプレートを用意し，サンプルとして 5 人の学生に同じ意味の手話動作を行ってもらった．動作を区切る閾値は±20°とし，それぞれ分割ごとにテンプレートとの比較を行った．また，両手の平均マッチ率を計算する際に，両手が動いているときは両手の平均を，片手のみが動いているときは動いていない手は考慮せずに，動いている片手のみの数値のマッチ率を計算した．その結果の一例を表 3.1.1，3.1.2，3.1.3 に示す．

表 3.1.1 手話動作「おはようございます」の実験結果

「おはようございます」の両手平均マッチ率(%)			
	1分割目	2分割目	3分割目
朝	100	26.7	33.3
昼	46.2	33.3	66.7
あいさつ	33.3	54.5	**100**
初めて	26.7	100	**54.5**
また1	61.5	50	66.7
また2	80	26.7	33.3
会う	53.3	71.4	**62.95**

表 3.1.2 手話動作「こんにちは」の実験結果

「こんにちは」の両手平均マッチ率(%)			
	1分割目	2分割目	3分割目
朝	46.2	71.4	33.3
昼	100	72.7	66.7
あいさつ	66.7	40	100
初めて	33.3	30.8	**54.5**
また1	40	36.4	66.7
また2	80	26.7	33.3
会う	33.3	30.8	**62.95**

表 3.1.3 手話動作「また会いましょう」の実験結果

「また会いましょう」の両手平均マッチ率(%)				
	1分割目	2分割目	3分割目	4分割目
朝	61.5	80	71.4	71.4
昼	40	33.3	33.3	33.3
あいさつ	66.7	54.5	**52.25**	**68.9**
初めて	50	42.9	**69.05**	**69.05**
また1	100	83.3	50	83.3
また2	83.3	100	71.4	71.4
会う	83.3	71.4	**63.1**	91.65

マッチ率が 80%以上のものは赤，両手が同時に動いているものは灰色と太字で示す．

「おはようございます」は「朝」→「あいさつ」の流れで手話表現を行うため，最初に「朝」，その次に「あいさつ」の数値が高くなることが理想である．同様に「こんにちは」も，「昼」→「あいさつ」の流れで数値が高くなることが理想である．また，「また会いましょう」に含まれる単語「また」は，提案手法の動作の区切り方だと 2 動作に分割される．そのため，動作に「また」の 1 動作目(また 1)→「また」の 2 動作目(また 2)→「会う」の順に数値が高くなることが理想の結果と言える．実際の実験結果を見ると，「こんにちは」は理想の結果と言えるが，「おはようございます」や「また会いましょう」は，該当する単語以外にも高い数値が出てしまう，単語と単語の間の動作が他の単語のテンプレートとマッチしてしまい，ノイズとなってしまう事象が起きてしまった．

次に，N-gram で算出した共起語と共起頻度の結果の一例を表 3.1.4，表 3.1.5，表 3.1.6 に示す．

表 3.1.4 「ご飯」の共起語上位 10 単語と共起頻度

「ご飯」の共起語			
単語	共起頻度	単語	共起頻度
食べる	0.337182	ー	0.041570
朝	0.073903	てる	0.041570
作る	0.053117	する	0.039260
晩	0.053117	昼	0.027713
夜	0.046189	お昼	0.027713

表 3.1.5 「お腹」の共起語上位 10 単語と共起頻度

「お腹」の共起語			
単語	共起頻度	単語	共起頻度
いっぱい	0.291666	ー	0.041666
すく	0.097222	ん	0.041666
空く	0.097222	ちゃう	0.034722
なる	0.090277	する	0.034722
食べる	0.048611	何	0.034722

表 3.1.6 「食べる」の共起語上位 10 単語と共起頻度

「食べる」の共起語			
単語	共起頻度	単語	共起頻度
てる	0.081327	られる	0.036483
ん	0.047884	れる	0.033696
何	0.037750	の	0.033189
ご飯	0.036736	もの	0.032936
ー	0.036483	ちゃう	0.029896

「ご飯」は「食べる」が一番共起しやすく，他にも「朝」，「夜」などの結果が出力された．また，「お腹」は「いっぱい」，「空く」，「食べる」などが共起しやすく，「食べる」は「ご飯」，「もの」などが共起しやすい結果と出力された．これらの結果はそれぞれの単語に共起することは自然と考えられる．しかし，「ー(長音符)」や「ん」などの不自然な結果も出力された．

3.2 考察

手話表現の動作判別については，理想に近い状態が出力された．しかし，理想から遠ざける結果も出力されている．これは手話の一動作が終わり，次の動作に移る際の手の移動を，手話の一動作と同じ表現になってしまうことが原因と考えられる．この問題を解決するために，動作の分割やマッチ率の閾値の調整が必要であると考えられる．

また，より精度を高めるために用いた n-gram での処理の結果は，MeCab での形態素解析がうまく行えていない部分が見受けられた．これは用いたコーパスに方言などの未知語が多いことが原因と考えられる．そのため，形態素解析された結果から名詞と判断されている記号や不自然な単語を取り除くことが必要だと思われる．さらに n-gram 用のコーパスの量が少ないため，コーパスの語彙数を増やすことにより，精度を高める結果が求められると考えられる．

4. まとめ

4.1 まとめ

手話を利用している聴覚障害者と健常者はコミュニケーションを取る際に，健常者に手話の知識がなければコミュニケーションを取るだけでも手間がかかり，ストレスを感じやすいのが現状である．そのような問題を解決するために，手話翻訳のために単眼カメラから取り込んだ動画をテンプレートと比較し，さらに精度を高めるために言語モデルを用いた実験を行った．取り込んだ動画を動作ごとに分割しベクトル化を行う．単語を最小単位に分割したテンプレートとのマッチ率の計算に最長共通部分列を利用し，入力された動画とテンプレートとの類似度を求めた．また，精度を高めるための言語モデルは n-gram を利用し，単語ごとの共起頻度を算出した．その結果，手話動作の類似度比較については理想的な流れが出たが，ノイズが乗ってしまっている部分が見受けられた．原因は，手話の動作の合間に起こる手の移動が，用意したテンプレートと一致してしまったことである．この問題を解決するためには，動作の分割と，最長共通部分列で算出したマッチ率の閾値の調整が必要だと考えられる．また，n-gram で共起語を算出した結果，上位に不自然な単語が見受けられた．これは，用意したコーパスの量の少なさと形態素解析がうまく行えていないことが原因である．そのため，コーパスの語彙を増やすことや，形態素解析された結果を見直し，不要な語を取り除くことが必要である．

4.2 今後の展開

今後は挙がった問題点の解決のため，動画像処理の閾値の調整や，n-gram のためのコーパスを増やす他，今回提案したアルゴリズムを動画像処理部分，言語処理部分をつなげてシステム化を行う．また，現在のジェスチャ認識だけでの手話表現の区別には限界があるため，指先の認識を行うことにより，手話表現の区別の判断材料が増え，より精度の高い手話翻訳につながると考えられる．

参考文献

[大津 81] 大津 展之:“パターン認識における特徴抽出に関する数理的研究”，電子技術総合研究所研究報告， vol.818，1981．

[安達 00] 安達 久博:“手指動作記述文間の類似性に基づく類似の動作特徴を含む手話単語対の抽出方法”，自然言語処理 Vol.7 No.4，2000．

[安達 01] 安達 久博:“手指動作記述文間の類似性に基づく手話単語の分類方法”，自然言語処理 Vol.8 No.3，2001．

[吉田 08] 吉田 知訓，間瀬 心博，北村 泰彦:“質問応答 Web サイトからの関連語ネットワークの自動抽出”，電子情報通信学会技術研究報告．AI，人工知能と知識処理 108(119)，pp75-80，2008

[藤本 14]藤本 美加，井上 聡:“手話翻訳のための手形状認識アルゴリズム”，2014 年度人工知能学会全国大会，2C3-4，2014．