UNRESTRICTED VOICE ANNOTATIONS AND SEARCH OF PERSONAL PHOTOGRAPHS IN A MOBILE PHONE

Xavier Anguera, Mauro Cherubini and Nuria Oliver

Telefonica Research
Multimedia research group, Via Augusta 177,
08027, Barcelona, Spain
{xanguera, mauro, nuriao}@tid.es

ABSTRACT

Mobile phones have become truly multimedia devices. It is common to observe users capturing and consuming photos and videos on their mobile phones on a regular basis. As the amount of digital multimedia content expands, it becomes increasingly difficult to find specific images in the device. Therefore, novel mobile multimedia search applications and algorithms are needed. In previous work, we have presented MAMI (Multimodal Automatic Mobile Indexing), a light-weight mobile phone application that allows users to annotate, index and/or search for digital photos on their phones via a combination of speech, text or image input. When using speech annotations, MAMI uses a Dynamic Time Warping (DTW)-based pattern matching algorithm in order to find pictures that are annotated with a matching acoustic query. Such an approach is language and vocabulary independent, and light-weight-enough to be run in real-time on the mobile phone. The main drawback of using DTW is that only full acoustic sequences can be matched, therefore constraining the type of acoustic annotations that can be used for tagging and search. In this paper we expand the acoustic search and retrieval capabilities of MAMI by enabling *unrestricted acoustic sentence matching*. We propose substituting DTW by the recently proposed Unbounded Dynamic Time Warping (U-DTW). Given a spoken query, U-DTW finds all database acoustic annotations with matching acoustic segments, regardless of their start-end points and length. In addition, we propose a number of speed-ups to U-DTW that make it suitable for mobile applications like MAMI.

Index Terms**—** Dynamic time warping, partial sequence match, pattern matching, mobile applications, multimedia indexing

1. INTRODUCTION

Mobile phones have become multimedia devices. It is common to observe users capturing photos and videos on their mobile phones, instead of using digital cameras or camcorders. As consumers generate an increasing number of digital multimedia content, its organization, search and retrieval becomes a non-trivial task. Often, this rich multimedia content is lost in the users' personal repositories due to the lack of efficient and effective tools for tagging and searching the content. One solution to this multimedia data management problem is the addition of textual annotations or metadata to the content [1, 2, 3, 4], therefore allowing users to search for multimedia information using keywords related to their annotations. However, most of the image tagging systems proposed to date are designed to add the metadata at a *later time* and on a *desktop* computer. Time lag, and device and context change significantly reduce the likelihood that users will perform the task, and their accurate recall of the context in which a particular photo or video was taken.

Recently, there has been some research directed towards real-time multimodal annotations on mobile phones. Related work takes advantage of GPS-derived location metadata [5] or content-based image retrieval and user verification to achieve high-level metadata [6]. Speech tags are recognized in [2] by means of a server connected to the mobile phone. An important drawback of server-based systems is the need of reliable connectivity to the server in order to work. In recent work [7, 8], we have proposed and experimentally evaluated a mobile prototype named MAMI (Multimodal Automatic Mobile Indexing) which is able to annotate and retrieve user photographs on the phone using any combination of visual descriptors and speech and textual tags. MAMI is a stand alone application, *i.e* does not need any connectivity to a server, and works in real-time thanks to fast indexing and search algorithms.

MAMI's approach to retrieve images with certain acoustic tags is based on DTW [9], *i.e.*, it is a pattern matching approach. Using DTW has several advantages: a) it is light-weight and hence feasible to run in any cell phone; b) it is language independent; and c) the vocabulary of the usable tags is not restricted. Traditionally, DTW has been used for many applications including keyword recognition [10], template-based speech recognition [11] and music synchronization [12]. In MAMI, DTW is useful to compute the distance between acoustic tags reasonably fast, but does not allow for matching start-end points to be elsewhere than the start-end positions of both tags. Hence, MAMI's acoustic annotations are limited to be single word tags (or multiple words in a single, indivisible sequence). During user studies conducted with MAMI [8], participants frequently reported the need to tag pictures with full sentences (not only with 'atomic' keywords) and be able to retrieve them using fragments of the original sentence.

Inspired by the modifications to DTW by [13, 14, 12], we have recently proposed a variation of DTW named Unbounded-DTW (U-DTW) [15] which improves the state-of-the-art in speed and increased flexibility of start-end locations in the matching sequences of both compared signals. In this paper, we have implemented U-DTW in the MAMI prototype – replacing the original DTW algorithm – in order to allow for acoustic annotations that are full sentences.

In the old prototype, the user was forced to recall the exact acoustic tag that was used during the tagging phase to retrieve the pictures. With the U-DTW algorithm the user is able to use a fraction of the original acoustic tag to retrieve the picture s/he is interested into.

During the development of this work X. Anguera is partially funded by the Torres Quevedo program

Note that speed is a key factor in the matching process, particularly given that the algorithm runs on a mobile device. Therefore, in this paper we propose several modifications to U-DTW that significantly increase its speed. We prove the feasibility of U-DTW – both in terms of accuracy and speed – with a database generated from data recorded with MAMI.

This paper is structured as follows: Section 2 briefly overviews the MAMI prototype and Section 3 reviews the U-DTW algorithm and describes the newly proposed speedups. Our experiments are summarized in Section 4, followed by our conclusions in Section 5.

2. SYSTEM DESCRIPTION

Recently, we have proposed MAMI (Multimodal Automatic Mobile Indexing) [7], a multi-modal mobile application that allows users to annotate and search their pictures on the phone. MAMI's users can tag a photograph at the time of capture and later search for it using speech, text or image input. MAMI is implemented as a Windows Mobile application. In our experiments, we have used the HTC Touch^{TM} phone running Windows Mobile 6.0.

MAMI has two modes of use as depicted in Figure 1. Figure 1a) illustrates an example of MAMI's capture and annotate functionality. This mode implements the picture taking functionality and addition of metadata at the time of capture: time and date, location, user identity, speech annotation(s), text tag(s) and image-based features. When the user takes a digital photo or video with his/her mobile phone (step 1 on the Figure), this component automatically gathers available contextual metadata (step 2) and allows users to enter audio/text annotation(s) at the point of capture (step 3). The digital content and annotations are stored in MAMI's picture and metadata databases, which are locally stored on the phone (step 4). Note that a push-to-talk method is used for audio annotations in order to ensure that the system works in all acoustic conditions.

The second mode of use allows users to search and retrieve photos from their personal repository by means of speech, text or image queries (step 1 on Figure 1b)). In the case of speech queries, MAMI uses pattern matching algorithms (DTW in the original version, U-DTW in the version described in this paper) to compute the distance between the input speech and all existing audio annotations in the user's digital media database (steps 2 and 3). MAMI returns the N (where N is typically 4) photos whose annotations are the closest to the input query (step 4), and presents them to the user (step 5). All the processing is carried out on the phone.

Figure 2 shows some exemplary screenshots of the MAMI's prototype. Figure 2a) shows the interface presented to the user after a photograph is taken to annotate it with text and/or audio tags. Figure 2b) depicts the search interface, which adds a photo query capability to the audio and text input. Figure 2c) displays the results of a search, showing the 9-best matching results. Any of the thumbnails can be clicked and expanded to full screen as seen in Figure 2d).

2.1. Multimodal search and retrieval

Upon taking a photograph, MAMI automatically stores contextual metadata that is gathered from the system and the environment, such as time/date, cellID (which is useful for geo-tagging and available in all phones) and the identity of the user taking the photo. MAMI then prompts the user with an annotation screen shown in Figure 2a) where one or several text and speech annotations can be entered. Furthermore, each image is analyzed in the background and its features are stored to allow users to search for images by means of image queries. Next, we review how each input modality is dealt with in the MAMI prototype.

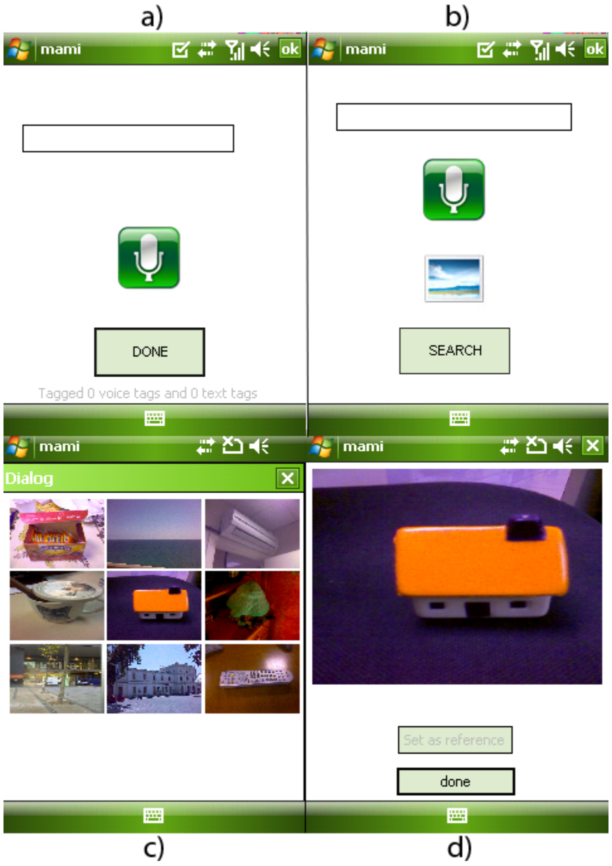


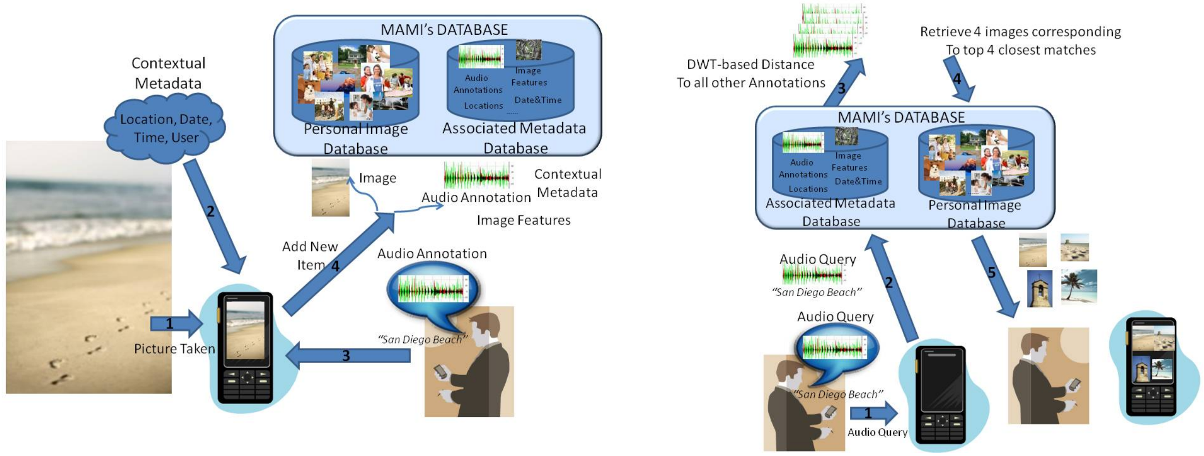
Fig. 2. MAMI screenshots: a) Annotation, b) Search, c) Search results, and d) Zoom on result.

2.1.1. *Text-based annotations*

Text strings can be entered in the annotation screen and are stored in the metadata file associated with each photograph. Any number of free-text tags may be entered for each photograph. In search mode, the input text string is compared to each of the stored tags via dynamic programming and using the Edit Distance between characters, such that the optimal alignment between the query and reference textual tags is found by considering insertions and deletions of characters. The final distance corresponds to the number of aligned matching characters, normalized by the total number of characters in both sequences.

2.1.2. *Audio-based annotations*

Audio input is used in the MAMI prototype either for tagging or searching. In both cases, the audio recording is carried out via a push-to-talk method. The audio tags are stored in disk as .wav files and are converted to 10 Mel Frequency Cepstral Coefficients (MFCC) [16] extracted every $20ms$, including CMN (Cepstral Mean Normalization) and excluding the C0 component. This choice of acoustic features was designed to optimize the discriminative power of the audio descriptor, while keeping the feature extraction as fast



(a) (b)

Fig. 1. MAMI's modes of operation: a) capture and annotation mode b) search mode.

as possible ($10ms$ samples yield slightly better search performance but are much slower to process). The obtained feature vectors are stored in memory for later use.

In order to compare two audio tags, in previous work we applied a threshold-based speech activity detector (SAD) to both signals to eliminate start-end silences, and then used the Dynamic Time Warping (DTW) algorithm (using Euclidean distance between speech vectors) to find the optimum alignment between the signals, and therefore a matching score for each pair. In the work presented in this paper, we have eliminated the need of SAD and replaced the DTW algorithm for the recently proposed Unbounded DTW (U-DTW) algorithm. U-DTW is a fast subsequence matching algorithm that returns the locations in both signals of found matching segments, together with their matching score. A minimum matching length of $0.4s$ is enforced to reduce false alarms. The algorithm is described in more detail in section 3.

Once the input query has been compared with all acoustic tags in the user's database, the 9-best photographs are shown to the user in an interface shown in Figure 2c).

2.1.3. *Visual-based search and retrieval*

Visual descriptors are computed for each taken photograph and are compared with those of a query photograph in order to find images in the database that are similar to the image query.

Edge-derived features have traditionally been an important and computationally light-weight approach to characterize image content. MAMI's image processing module uses the Edge Histogram Descriptor (EHD) of the MPEG-7 Visual Standard for measuring similarity between images. The EHD is designed to capture the spatial distribution of edges in an image by computing a histogram that represents the frequency and the directionality of the brightness changes in the image [17].

To extract the EHD, a given image is first sub-divided into 4×4 sub-images. Each sub-image is further divided into non-overlapping image blocks, which are the basic units for edge extraction. The number of image blocks is typically fixed (*e.g.* 1100). Therefore, the block size depends on the resolution of the image. Each of the image blocks is then classified into one of the five types of edges, namely: vertical, horizontal, 45-degree diagonal, 135-degree diagonal or non-directional. Then a 5-bin histogram is constructed to characterize the distribution of edges in considered the sub-image. The end of this process yields an edge histogram with a total of 80 (16×5) bins, since there are 16 (4×4) sub-images. This 80 dimensional histogram constitutes the image feature vector that is stored with the rest of the image metadata for later use.

In order to compare two images, the Euclidean distance is applied to their image feature vectors, such that the smaller the distance, the more similar the images are. When the user looks for a specific image via an image sample, the MAMI prototype computes the Euclidean distance between the input image's visual feature vector and all stored image feature vectors.

2.1.4. *Multimodal Disambiguation*

MAMI's queries may be performed by using any of the three modalities previously explained. In [18], we have proposed an algorithm that, given a monomodal query (audio or image), leverages the multimodal information associated with each picture in order to select an optimum subset of images to return to the user.

3. SPEECH ANNOTATION SEARCH AND RETRIEVAL

The main contribution of this paper is the implementation of U-DTW in the MAMI prototype (instead of the standard DTW), in order to handle matching of unconstrained voice annotations. In this section we review the U-DTW algorithm – initially proposed in [15] – and describe a number of novel speedup improvements to U-DTW to make the algorithm suitable for small processors such as those in mobile phones.

3.1. Unbounded Dynamic Time Warping Algorithm

The U-DTW algorithm has two main properties that make it very suitable for the task at hand. First, it does not pose any restrictions on the start-end positions of the two audio segments that are to be matched. Therefore, any matching subsequence in both the query and the database segments can be found. Second, it is very efficient as it does not compute every possible frame-pair distance between the two sequences that are compared (as it is the case in standard DTW). Hence, U-DTW is much faster than other recently proposed

DTW alternatives [13] and [14] while maintaining high accuracy levels.

In order to restrict the number of resulting matches, two parameters need to be set in the algorithm. First, a *minimum length* L_{min} is used to only consider matching subsequences that are both longer than the minimum time (typically set around 500ms, which corresponds to approximately a 2 syllable word). Second, a *maximum time warping* of $2X$ (and minimum of $\frac{1}{2}X$) between consecutive frames is enforced by defining proper local constraints.

3.1.1. Local Constraints

In all steps of the U-DTW algorithm, we constraint the number of possible frame jumps (*i.e.* local constraints) to those in Figure 3. As we will see next, the jumps in (a) correspond to the forward path of the algorithm, and the ones in (b) to the backward path. Note how these local constraints differ from the standard DTW in that neither strict insertion nor strict deletion steps are allowed, which allows at most $2X$ and $\frac{1}{2}X$ warping of one signal to the other. This limitation is realistic in the case of spontaneous speech and is very useful to avoid long consecutive insertions/deletions given that no global constraints are applied.

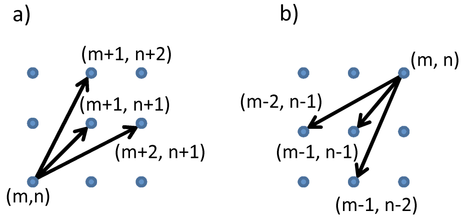


Fig. 3. Allowed frame jumps in U-DTW.

3.1.2. Algorithm Description

The algorithm works as follows: Given two acoustic sequences X_U and X_V and their acoustic feature sequences given by $U := (u_1, u_2, \ldots, u_M)$ and $V := (v_1, v_2, \ldots, v_N)$, we define the *similarity* $\mathcal{S}(m,n)$ between any two feature vectors, u_m and v_n, with $m\epsilon[1 : M]$ and $n\epsilon[1 : N]$, as their normalized inner product (or cosine of the angle θ between the vectors): $\mathcal{S}(m,n) = \cos\theta = \frac{<u_m,v_n>}{\|u_m\|\|v_n\|}$. Hence, the similarity values are bounded between -1 and 1.

Additionally, two support matrices are defined: (1) a global similarity matrix $D_g(m,n)$ that contains the optimum path accumulated similarity at each location (m,n); and (2) a matrix $M(m,n)$ that keeps the length of the optimal paths up to each location (m,n). All these matrices are set to all zeros at startup. The pseudo-code of the U-DTW algorithm is given by Algorithm 1.

Unlike other DTW-based algorithms, the similarities $\mathcal{S}(m,n)$ are computed only when needed, according to the forward-backward path finding algorithm, which brings significant computational savings.

3.1.3. Synchronization Points

One of the key steps in the U-DTW algorithm is the proper selection of the *synchronization points* (SPs) which the forward path algorithm uses as starting points to look for matching segments. Given that starting and ending points for a match can occur anywhere within the two sequences a naive approach would be to check for matches at every single location. The U-DTW algorithm takes advantage of the fact that only matches with minimum length L_{min} are considered in order to compute distances in a sparser set of points (the SPs) instead of all points, while ensuring that, if there is a matching segment, it will be found.

Algorithm 1 Unbounded-DTW

```
Define appropriate synchronization points (see Section 3.1.3) in
locations $(m,n)$. Compute the similarities for these points and
assign them to $\mathcal{S}(m,n)$ and $D_g(m,n)$ and set $M(m,n) = 1$.
for $m = 1 to M$ do
  for $n = 1 to N$ do
    if $M(m,n) \neq 0$ (a synch point or a possible path) then
      Apply forward path alg. (see Section 3.1.4).
    end if
  end for
end for
for all paths found in the forward pass do
  Apply backward path alg. (see Section 3.1.4).
  if Resulting Path length $> L_{min}$ then
    Register found path's start-end points and score
  end if
end for
```

Both the accuracy and speed of the algorithm depend on the accurate selection of SP: sparse SP locations increase the processing speed at the expense of possibly missing matching segments, whereas dense SPs are computationally more expensive to process. There are several possible ways to define the SP's, refer to [15] for a complete explanation. In this paper we use the horizontal synchronization bands, which define SP's along the x axis for every $y = \tau_h k$ with $k = 0 \ldots \frac{N}{\tau_h}$ where N is the length of the sequence in the vertical axis, and τ_h is the defined vertical separation between bands, with theoretical maximum value $\tau_h^{max} = \frac{L_{min}}{2}$.

3.1.4. Forward-Backward Paths Algorithm

As seen in Alg. 1 given every SP the algorithm first searches for possible matching sequences using a forward path algorithm, this finds the possible match ending points. Then, for each possible match it applies a backward path algorithm from the initial SP to find the possible starting points. Finally, if the resulting sequences from both segments are at least L_{min} long, their score is computed and are returned as matches.

In particular, for any considered frame-pair location (m,n), the forward and backward path algorithms check whether the current path can be extended to any of the surrounding frame-pair locations, conditioned to the local constrains seen in Figure 3.

Given $(m',n') = (m,n) + (i,j)$ where
$(i,j)\epsilon\{(1,1),(1,2),(2,1)\}$ for the forward path, and
$(i,j)\epsilon\{(-1,-1),(-1,-2),(-2,-1)\}$ for the backward path, a new frame-pair position (m',n') is added to the currently considered path if the following conditions are met:

- The normalized global similarity score of the current path is greater than any previous paths (if any) at that location:

$$\frac{D_g(m,n) + \mathcal{S}(m',n')}{M(m,n)+1} > \frac{G_g(m',n')}{M(m',n')} \tag{1}$$

- The normalized global similarity is greater than a predefined cutoff threshold.

$$\frac{D_g(m,n)+\mathcal{S}(m',n')}{M(m,n)+1} > Thr \tag{2}$$

If successful, we set: $M(m',n') = M(m,n)+1$ and $D_g(m',n') = D_g(m,n) + \mathcal{S}(m',n')$. Note how Eq. 1 allows us to obtain the optimum DTW path without the need to backtrack, which is the key to finding optimum alignments while avoiding the pre-computation of the entire similarity matrix $\mathcal{S}(m,n)$. Also note how at any given frame-pair location, the path can branch out in as many as 3 independent paths.

Any path is terminated at location (m,n) when none of the possible (m',n') meet the conditions above. If we are in the forward path step, we find the starting SP and execute from there the backward path algorithm to find the starting points. Finally, we keep the longest of all paths that pass through the considered SP, returning the total path (backwards + forward) and its average score if it exceeds the minimum length L_{min} in both dimensions.

3.1.5. *Optimizations to U-DTW*

In order to successfully run the U-DTW algorithm on a mobile phone (with limited computing capabilities), the computation requirements of the algorithm need to be minimized while maintaining the accuracy of retrieval. In this section, we present three optimizations to the original U-DTW algorithm and we validate them in the experimental section.

The first optimization consists on the elimination of the diagonal path (*i.e* $(m',n') = (m+1,n+1)$ and $(m',n') = (m-1,n-1)$) from the local constraints. Although deviating from the standard definition of DTW, the remaining local constraints allow us to consider all points that were before visited by the diagonal constraint.

The second optimization concerns the selection of the SP as possible starting points for the matching paths. When the normalized dot product computed in these positions is very low, it is highly improvable that matching segments will start from that point. Hence, they are ignored when $\mathcal{S}(m,n) < 0$. Note that these points could still be considered further on by pre-existing paths.

The third optimization expands on the second optimization by also eliminating the SP points immediately surrounding the eliminated SP, regardless of their value $\mathcal{S}(m,n)$. This is done considering that speech is a monotonous signal, therefore there should not be big changes between adjacent distances.

4. EXPERIMENTS

In this section, we demonstrate the feasibility of using U-DTW to perform full-sentence acoustic search locally on a mobile phone within MAMI. While the U-DTW algorithm has been fully implemented in MAMI, we performed all experiments on a PC at 2.4GHz running Ubuntu for convenience. We compare – both in terms of accuracy and processing time – the proposed U-DTW approach and its optimizations with the standard DTW algorithm.

4.1. Database and Metrics

The database used for the experiments was recorded in-house by 23 people using an HTC-touch cell phone in a variety of office background conditions [7]. Each person recorded a total of 47 isolated words, each one repeated 5 non consecutive times, to a total of 235 recorded words per person. From these words in this analysis we used 24 different words with 5 repetitions each, therefore a total of 120 words per person.

Algorithm	accuracy	Proc. time	ratio
Classic DTW	97.19%	1.1ms	1
U-DTW (10ms)	94.05%	25.37ms	0.54
U-DTW (20ms)	91.57%	3.89ms	0.72

Table 1. Performance evaluation of DTW versus U-DTW

All files were stored at a sampling rate of 11.025Hz with 16bit/sample. Each file was parameterized with 10 MFCC every 10ms or 20ms, and Cepstral Mean Substraction (CMS) was applied to the final features. A simple energy-based voice activity detector (VAD) was used to eliminate starting and ending non-speech regions. In order to add context to the words, two different starting and two different ending short sentences of 0.5s to 1.8s were recorded by a single speaker. Acoustic test sequences X_U and X_V were built by appending such segments to each recorded word: $X_U[i] = start_1 + word_i + end_1$ and $X_U[j] = start_2 + word_j + end_2$.

Tests were performed in the following way: For each acoustic sequence $X_U[i]$ of each speaker, the best matching segment score was found with each of the acoustic sequences $X_V[j]$ by the same speaker given that $i \neq j$, totalling $328,440$ matching pairs. Note that both $X_U[i] - X_V[j]$ and $X_U[j] - X_V[i]$ comparisons were computed in order to measure whether any asymmetry in the algorithm could affect the final results.

The main metric used is the matching accuracy, computed in the following manner: Given a comparison on two acoustic sequences $X_U[i]$ and $X_V[j]$, for all sequences i in $X_U[i]$ ($X_U[i]$ used as query) we compute the percentage of times that the best matching word in $X_V[j]$ corresponds to a different iteration of the same word. The same is done for each sequence j in $X_V[j]$ ($X_V[j]$ used as query) and the average across all words and speakers is computed. Other two metrics considered are the average processing time per sequence-pair (excluding the parametrization step) and the average ratio of computed frame-pair distances in the sequence-pair similarity matrix, as indicators of algorithmic efficiency/speed.

4.2. Results

Table 1 shows results comparing the standard DTW and the proposed U-DTW algorithms. Results for DTW are computed using a frame rate of 20ms/frame and considering that we know beforehand the optimum start-end points of matching words in the database, which is not feasible in a real application but serves in here to establish a performance ceiling. As we can see, performance and running time using standard DTW is very competitive and clearly outperforms U-DTW, although it does not allow MAMI users to store and search for unrestricted sentences. The table summarizes results for U-DTW both with 10ms and 20ms frame rates. Although 10ms obtains better accuracy, it is very slow to run and therefore not usable in MAMI. Note how U-DTW with 20ms frame rate computes in average 76% of distance-pairs between both compared sequences, which will be reduced using the proposed optimizations.

One important parameter in setting up the U-DTW algorithm is the minimum cutoff threshold applied to both forward and backward paths, as it determines how quickly possible matching paths are pruned and indirectly affects the accuracy. Figure 4 shows the accuracy, the average ratio of computed frames and the average pair-

optimization	accuracy	Proc. time	ratio
Original U-DTW	91.57%	3.89ms	0.71
Optim. 1	82.22%	2.13ms	0.44
Optim. 2	87.68%	2.28ms	0.43
Optim. 3	87.73%	2.20ms	0.39

Table 2. Performance of the proposed optimizations to U-DTW

wise processing time (excluding parametrization) for U-DTW with 20ms per frame with respect to such cutoff threshold. Note how accuracy finds a broad plateau from 0.3 to 0.7, with an optimum value at 0.4. On the other hand, the ratio of computed frame-pairs seems very correlated with the average time spend in each sequence-pair comparison. Given these results, it makes sense to optimize performance by reducing the number of frame-pairs being computed in the algorithm as evaluated below.

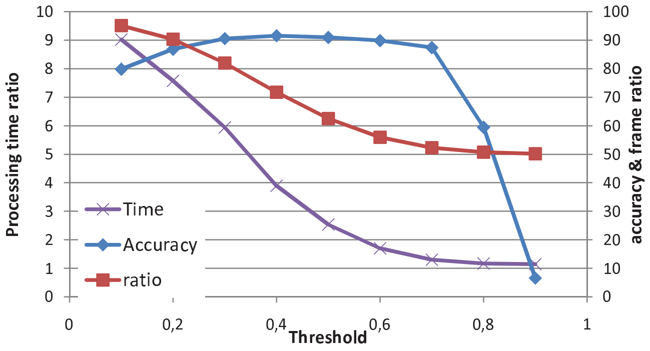


Fig. 4. Selection of cutoff threshold dependent on accuracy, time and frame ratio

Finally, Table 2 shows the results of applying the different proposed optimization techniques to the original U-DTW algorithm (shown in the first row as reference). In all cases accuracy drops while processing time gets reduced. Given that in MAMI the 9-best matching images are returned to the user it is not very crucial to obtain optimum performance, while it is necessary that response time be minimal as the user will be spending all this time waiting for results. Optimization 1 achieves pretty good processing times but its accuracy drops to almost 10% absolute below the original system, which could be already noticeable by the user. Between optimization 2 and 3 accuracies are very similar while optimization 3 achieves better processing time and bigger reduction of the ratio of processed frame distances. These values are double the time spent by classic DTW in processing the same queries. However it is important to notice that the DTW algorithm does not allow the subsequence matching functionality.

5. CONCLUSIONS AND FUTURE WORK

As mobile phones have evolved over the years, they have become people's multimedia companions, being used, among other things, to take and manage personal pictures. In recent work, we have developed MAMI, a multi-modal mobile phone prototype that can be used to annotate user's pictures at the time of capture and to search and retrieve desired pictures when needed. MAMI allows for audio, text and image annotations. In the initial prototype, speech annotations are limited to individual acoustic sequences that are searched for via pattern matching over the whole sequence. In this paper, we have proposed to use Unrestricted Dynamic Time Warping (U-DTW) to search for matching acoustic subsequences between query and reference annotations. In the case of MAMI, a very important performance metric is response time. Therefore, we also propose three optimizations to make U-DTW suitable for multimedia applications on mobile phones. Future work includes a user study with the new functionalities of MAMI.

6. REFERENCES

[1] M. Davis, *Readings in Human-Computer Interaction: Toward the Year 2000*, chapter Media Streams: An Iconic Visual Language for Video Representation, pp. 854–866, Morgan Kaufmann, 1995.

[2] T. Hazen, B. Sherry, and M. Adler, "Speech-based annotation and retrieval of digital photographs," in *Proc. of INTERSPEECH*, 2007.

[3] K. Roden and K.R. Wood, "How do people manage their digital photographs?," in *Proc. CHI'03*, ACM Press, Ed., 2003, pp. 409–416.

[4] P. Yee, K. Swearingen, K. Li, and M. Hearst, "Faceted metadata for image search and browsing," in *Proc. of CHI 2003*, ACM Press, Ed., 2003.

[5] K. Toyama, R. Logan, A. Roseway, and P. Anandan, "Geographic location tags on digital images," in *Proc. of Intl. Conf. on Multimedia*, ACM Press, Ed., 2003.

[6] L. Wenyin, S.T. Dumais, Y.F. Sun, and H.J. Zhang, "Semi-automatic image annotation," in *Proc. of INTERACT'01*, 2001.

[7] Xavier Anguera and Nuria Oliver, "MAMI: Multimodal annotations on a mobile phone," in *Proc. of MobileHCI'08*, 2008.

[8] Mauro Cherubini, Xavier Anguera, Nuria Oliver, and Rodrigo de Oliveira, "Text versus speech: A comparison of tagging input modalities for camera phones," in *Proc. MobileHCI*, 2009.

[9] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 26, pp. 43–49, 1978.

[10] Alan L. Higgins and Robert E. Wohlford, "Keyword recognition using template concatenation," in *In Proc. ICASSP'85*, 1985.

[11] Mathias De Wachter, Mike Matton, Kris Demuynck, Patrick Wambacq, Ronald Cools, and Dirk Van Compernolle, "Template-based continuous speech recognition," *IEEE Transactions TASLP*, vol. 15, no. 4, pp. 1377–1390, May 2007.

[12] Meinard Müller, *Information Retrieval for Music and Motion*, Springer, New York, USA, 2007.

[13] Alex Park and James R. Glass, "Towards unsupervised pattern discovery in speech," in *In Proc. ASRU'05*, Puerto Rico, 2005.

[14] Armando Muscariello, Guillaume Gravier, and Fredric Bimbot, "Audio keyword extraction by unsupervised word discovery," in *Proc. INTERSPEECH'09*, 2009.

[15] Xavier Anguera, Robert Macrae, and Nuria Oliver, "Partial sequence matching using an unbounded dynamic time warping algorithm," in *Proc. ICASSP'10 (to appear)*, 2010.

[16] P. Mermelstein, "Distance measures for speech recognition, psychological and instrumental," *Pattern Recognition and Artificial Intelligence*, pp. 378–388, 1976.

[17] Chee Sun Won, Dong Kwon Park, and Soo-Jun Park, "Efficient use of MPEG-7 edge histogram descriptor," *ETRI*, vol. 24, no. 1, pp. 23–30, February 2002.

[18] Xavier Anguera, JieJun Xu, and Nuria Oliver, "Multimodal photo annotation and retrieval on a mobile phone," in *Proc. MIR'08*. 2008, ACM Press.