# Word-level Stroke Trajectory Recovery for Handwriting with Gaussian Dynamic Time Warping

Anonymous

Anonymous

**Abstract.** Handwriting trajectory recovery has recently gained more attention for practical applications such as personalized messages. It is a sequence learning problem from image to handwriting stroke sequence where Dynamic Time Warping (DTW) is a preferred loss function. However, aligning two varying length sequences in DTW loss accumulates the differences of predicted and ground truth strokes for the entire line-level text. As a result, averaging over long sequences in DTW loss, it cannot distinguish between a small number of perceptually significant errors and a large number of visually insignificant errors. To address this issue, we propose two new strategies. First, we propose applying DTW to words instead of line-level text so that the DTW loss for all the words in the line-level text is not averaged out. Moreover, for aligning the predicted and ground-truth sequences for each word, we propose to weight the cost matrix with a Gaussian function so that the far-off predicted strokes from ground truth are penalized heavily. This strategy for word-level stroke trajectory learning improves quantitative and qualitative results.

## 1 Introduction

Everyday, we capture a lot of handwritten information on digital device via stylus pen or on physical paper. Handwriting inscribed on the digital device is termed as online handwriting whereas handwriting on physical medium such as paper or scanned document is referred as offline handwriting [9, 15]. The advancement in the use of digital devices provide handful amount of online handwriting datasets [9, 2, 11], but offline handwriting is still more ubiquitous. Handwriting stroke Trajectory recovery is easier on online handwriting because of the availability of stroke order, velocity, but its becomes challenges task for unconstrained offline handwriting [18, 13]. Despite the difficulty of the task, handwriting stroke trajectory recovery from offline handwriting is of utmost importance to revolutionize the emerging applications such as personalized message writing on letters or greeting cards, signature verification, and script handwriting learning.

The earliest work on handwriting stroke recovery started before the deep learning boom, which utilized handcrafted local and global features with the taxonomy of clues to recover the handwriting trajectory for each alphabet letter

[7, 19]. [1, 20] used a semantic rules-based approach for sub-words with a graph traversal to reconstruct stroke trajectory for handwriting recognition. Nevertheless, they considered only alphabets to learn the trajectory of the stroke. [16] recovered the trajectory information for handwriting by segmenting and dividing the words into a temporal sequence of strokes. Furthermore, [14] recovered the handwriting stroke trajectory by scanning the lines of the word and and reproduces the same movement as executed by the writer. [17] finds the multi-stroke trajectory by matching it with the writing paths of template strokes using dynamic programming. Above mentioned researches exhibit limited application for recently introduced handwriting datasets.

Stroke trajectory recovery has made progress towards more realistic and complex handwriting datasets using deep neural networks in recent years. [5] introduced the first trainable convolution network for stroke trajectory recovery. This LSTM architecture learns strokes with Euclidean distance loss, making it hard to apply on long words with multiple strokes. Moreover, [12] added a CNN before LSTM to recover the stroke trajectory of the handwriting in images. However, this work is limited to stroke learning for mathematical equations, and in the current form, it is not being applied to words in the English language.

The most recent work related to stroke trajectory recovery is presented by [3], where LSTM is trained with a Dynamic Time Waring (DTW) loss function. They also introduced adaptive ground truths to make stroke ordering more flexible during training. [13] employes an LSTM architecture with an attention layer and Gaussian Mixture Model (GMM) trained with cross-entropy loss, but it learns to encode only a single Japanese alphabet.

The proposed application of a Gaussian function is different from the one in [13], where the matching path computed by standard DTW is Gaussian weighted. Whereas in our approach a Gaussian weighted cost matrix is used to align the sequences. In other words, in our approach a Gaussian function is applied inside DTW warping computation, while in [13] a standard cost matrix based on Euclidean distance is used in DTW. As we will demonstrate, this difference has a significant impact on the quality of predictive strokes.

All these architectures either use line of text [3, 5] or alphabet letter [13, 20, 16], but to the best of our knowledge, the stroke trajectory recovery network for words has not yet been proposed.

Moreover, we propose to compute the warping path during the alignment of predicted and ground truth sequences in DTW with Gaussian weighting. In this way, we penalize the warping path heavily if the predicted stroke is far-off the ground truth stroke as it adds a perceptually significant error in stroke trajectory recovery. Whereas, the predicted stroke points in the close vicinity of the ground truth are perceptually indistinguishable from original strokes. The Gaussian function for DTW has been used for time series classification [8] based on the phase difference between two time series, but its potential advantage for stroke trajectory recovery has not been explored before. The main contributions of this work are as follows: 1) A word-level handwriting stroke trajectory recovery method is proposed. It estimates loss for each word rather than averaging DTW

loss over the entire line-level text. 2) To better match the human visual perception of handwriting, we employ a Gaussian weighted cost matrix in DTW to generate a loss function for deep learning. It allows our network to tolerate minor deviations in aligning the predicted and ground truth strokes while penalizing large, easily noticeable deviations. 3) Our quantitative and qualitative results demonstrate the superior performance of our approach in comparison to the state-of-the-art (SOTA).

We introduce the method in Sec. 2 and demonstrate the experimental results in Sec. 3.

## 2 Method

In our work, we introduced two levels of granularity to learn the stroke trajectory for handwriting, the first is dividing a line-level text into words, and the second is to use the Gaussian function to weigh the cost matrix in DTW loss for each word.

### 2.1 Word-level datasets

IAM-online datasets [10] consists of line-level text with stroke ground truth information. To the best of our knowledge, the previous researches [3] for handwriting stroke trajectory recovery considered the text lines as input. The disadvantage of using text lines is the averaging out of loss function for all the words in the line. However, some words have a structure that is harder to learn (such as *stage*) than the less complex words such a *the*. Therefore, in our work, we propose to break the text lines into words to calculate DTW loss for each word. For this purpose, the strokes are divided into words for train and test sets.

For this, we use a simple rule defined below. Let the stroke sequence $S$ be composed of strokes as $[s_1, s_2, s_3..., s_n]$. Stroke $s_{i+1}$ merges with the previous stroke $s_i$ if the following set of conditions are obeyed.

$$\begin{cases} M(s_{i+1}, s_i), & \text{if } \wedge(\mathrm{s}_{i+1}) \geq \vee(\mathrm{s}_i) \\ M(s_{i+1}, s_i), & \text{elif } (\vee(\mathrm{s}_i) - \wedge(\mathrm{s}_i + 1)) \geq th \\ Sep(s_{i+1}, s_i), & \text{otherwise} \end{cases} \tag{1}$$

Where the symbol $\wedge(s_{i+1})$ and $\vee(s_i)$ represents the minimum x-coordinate for stroke $s_{i+1}$ and the maximum x-coordinate for stroke $s_i$ respectively. $M$ and $Sep$ stand for merge or separate stroke function. We merge the strokes if the later stroke in $S$ has already started before ending the previous stroke or the distance between the two strokes is less than the threshold ($th$). The value of $th$ is different for each line. It is calculated based on the average stroke's spacing in each text line. Therefore it is based on handwriting style.

Figure 1 shows a reasonably separated words from line-level datasets into word level datasets.

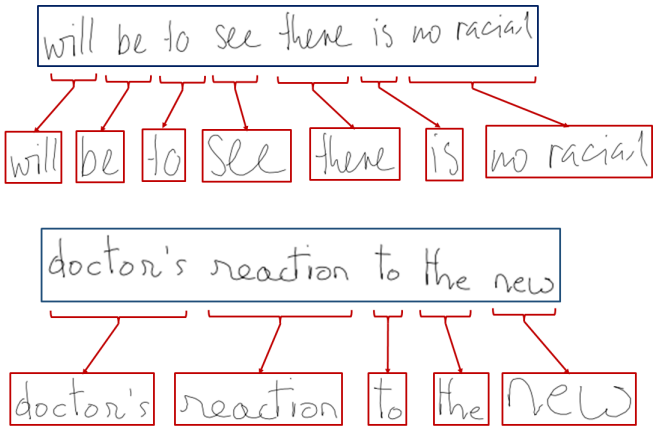


**Fig. 1.** Sample of the word-level IAM-online datasets we created.

### 2.2 Network architecture

Our architecture uses a CNN (seven convolutional blocks with ReLU) and LSTM layer. Convolutional filters have a 3x3 kernel size with 2x2 and 2x1 max pooling in each layer. Moreover, the input for the first convolutional block has a fixed height, and variable-width similar to [5, 3] in order to facilitate the processing of variable-length words for different handwriting styles. The block diagram of overall architecture is shown in Figure 2.

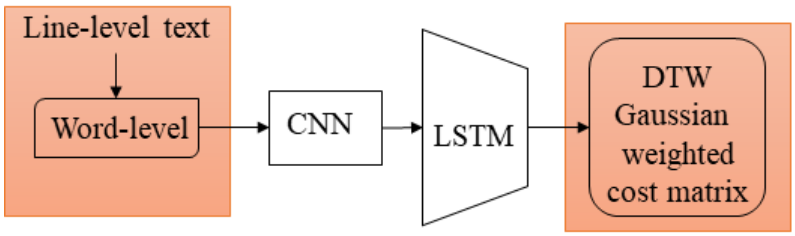


**Fig. 2.** The block diagram of our proposed architecture, the modules in orange highlight our contribution.

### 2.3 Loss function

In the next section, we introduce a Gaussian weighted cost matrix in DTW loss that emphasizes avoiding the costly alignment of far-off points in loss computation $\mathcal{L}_{DTW_{G_{align}}}$. Before explaining the Gaussian weighted cost matrix in DTW loss, we describe a simple Gaussian weighted DTW loss.

**Gaussian weighted DTW loss** In general, DTW [4, 6] computes the optimal match between GT ($T = (t_1, t_2, t_3, ....t_m)$) and predicted sequences ($P = (p_1, p_2, p_3, ....p_n)$) of different lengths by finding the warping path between two sequences. In DTW loss, the cost matrix is calculated as:

$$cost(i, j) = ||p_i - t_j||^2 \tag{2}$$

The accumulative cost matrix ($A$) as given as,

$$A(i, j) \;=\; cost(i, j) \;+\; min[A(i \;-\; 1, j), A(i \;-\; 1, j \;-\; 1), A(i, j \;-\; 1)] \tag{3}$$

for $1 \leq i \leq m$ and $1 \leq j \leq n$. Given matrix $A$, DTW computes the optimal warping path from $A(m, n)$ to $A(1, 1)$ as the alignment of points in $P$ to points in $T$ is expressed as index mapping $\alpha : \{1, \ldots, m\} \rightarrow \{1, \ldots, n\}$, where $\alpha$ is an onto function. The difference of aligned points is weighted by Gaussian function to compute DTW loss ($L_{DTW_{G_{Loss}}}$) as follows,

$$\mathcal{L}_{DTW_{G_{Loss}}}(P, T) = \sum_{i=1}^{m} G(||p_i - t_j||) \cdot ||p_i - t_{\alpha(i)}||. \tag{4}$$

The description of the Gaussian function $G(||p_i - t_j||)$ is same as described in the next section. The loss defined in (4) is used in [13].

**Gaussian weighted cost matrix in DTW** In previous works [6] and [3], the cumulative cost matrix $A$ at the $i^{th}$ stroke point of $P$ and $j^{th}$ stroke point of $T$ is calculated by their squared Euclidean distance. In our work, we propose to weight the distance ($||p_i - t_j||^2$) by Gaussian function $G$ as

$$G(||p_i - t_j||) \cdot ||p_i - t_j||^2. \tag{5}$$

To define $G$, we start with

$$H(||p_i - t_j||) = \sigma \left(1 - e^{-\left(\frac{||p_i - t_j||}{\sigma}\right)^2}\right), \tag{6}$$

where $\sigma$ is a constant related to Gaussian standard deviation. According to Eq. 6, $H$ ranges from 0 to $\sigma$. To define $G$, we clip the value of $H$ at 1 as follows:

$$G(x) = \begin{cases} H(x) & \text{if } H(x) > 1 \\ 1, & \text{else} \end{cases} \tag{7}$$

Fig. 3 shows the visualization of the Gaussian function $G$ used in our cost matrix. We evaluated our method for $\sigma = 2$ and $\sigma = 5$.

The Gaussian function $G$ directly affects the cumulative cost matrix $A$, where the $(i, j)^{th}$ entity of $A$ is given as:

$$A(i, j) = G(||p_i - t_j||) \cdot ||p_i - t_j||^2 + min[A(i-1, j), A(i-1, j-1), A(i, j-1)] \tag{8}$$

for $1 \leq i \leq m$ and $1 \leq j \leq n$. Given the cumulative cost matrix $A$, DTW computes the optimal warping path from $A(m, n)$ to $A(1, 1)$ as the alignment of points in $P$ to points in $T$ is expressed as index mapping $\alpha : \{1, \ldots, m\} \rightarrow \{1, \ldots, n\}$, where $\alpha$ is an onto function. Finally, the DTW loss given by the Gaussian weighted cost matrix in alignment $\alpha$ is given by:

$$\mathcal{L}_{DTW_{G_{Align}}}(P, T) = \sum_{i=1}^{m} ||p_i - t_{\alpha(i)}||. \tag{9}$$

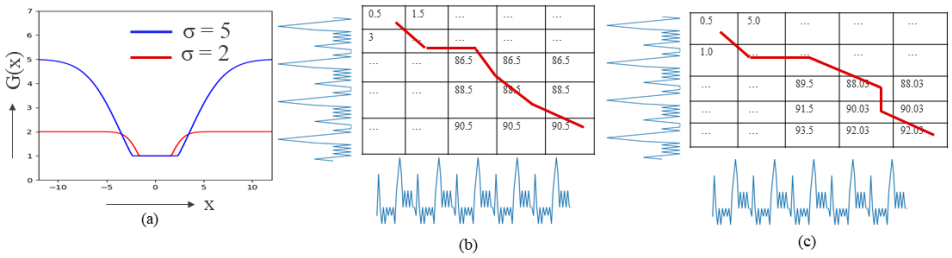


**Fig. 3.** (a) Gaussian function $G$ used to calculate the cost matrix for DTW alignment between GT ($T$) and predicted ($P$) strokes. (b) Warping path for Gaussian weighted cost matrix in DTW (red). (c) Warping path for standart DTW cost matrix. The x and y axes of cost matrix show the matched sequences.

Figure 3(a) shows the Gaussian function used in our work with $sigma = 2$ and $sigma = 5$. Figure 3(b,c) shows that there is a difference between the warping path based on the Gaussian weighted cost matrix and the standard DTW cost matrix.

The next section details the evaluation of handwriting stroke trajectory and the effect of the Gaussian function in cost matrix for strokes alignment.

## 3 Experimental evaluation

### 3.1 Data

In the IAM-online dataset [10], we have 10,927 annotations for line-level text with strokes information, out of which 7,402 are training, and 3,525 are testing text lines. After splitting text lines into words using the proposed word-level algorithm as described in Section 2.1, the size of training and testing datasets increase to 36,106 and 17,087 words, respectively. Figure 1 shows the sample images of the word-level dataset proposed in our work.

| Distance metric | | % $N_{t,p}$ | | $dist_{t,p}$ | | % $N_{p,t}$ | | $dist_{p,t}$ | |
|---|---|---|---|---|---|---|---|---|---|
| | | $T_0 = 5$ | $T_0 = 2$ | $T_0 = 5$ | $T_0 = 2$ | $T_0 = 5$ | $T_0 = 2$ | $T_0 = 5$ | $T_0 = 2$ |
| Line-level | | 0.0090 | 0.0215 | 0.4127 | 0.5021 | 0.0194 | 0.2802 | 0.0951 | 0.5868 |
| word-level | | 0.00487 | 0.0217 | 0.1758 | 0.3001 | 0.0108 | 0.1497 | 0.0395 | 0.724 |
| word-level | $\sigma = 2$ | **0.0035** | 0.0178 | 0.1563 | 0.2708 | 0.0049 | 0.1461 | 0.0120 | **0.2762** |
| $DTW_{G_{Loss}}$ | $\sigma = 5$ | 0.0036 | 0.0195 | **0.1301** | **0.2760** | 0.0115 | 0.1590 | 0.2780 | 0.6752 |
| word-level | $\sigma = 2$ | 0.00457 | 0.01829 | 0.184 | 0.278 | 0.0018 | 0.1442 | 0.0418 | 0.3109 |
| $DTW_{G_{Align}}$ | $\sigma = 5$ | 0.0040 | **0.0167** | 0.1828 | 0.2943 | **0.00098** | **0.1429** | **0.01128** | 0.4064 |

**Table 1.** The quantitative caparison of line-level, word-level, and word-level with Gaussian weighted DTW loss and Gaussian weighted cost matrix in DTW alignment

### 3.2 Evaluation Metrics

We use the same evaluation metric as [3]. It considers the percentage of predicted stroke points farther than $T_0$ pixels from their nearest GT denoted by $\%N_{t,p}$, and similarly, the percentage of GT stroke points father from their nearest predicted stroke denoted by $\%N_{p,t}$. The average distance of points in $\%N_{t,p}$ and $\%N_{p,t}$ is denoted by $dist_{t,p}$ and $dist_{p,t}$. To add the holistic view of GT and predicted sequence matching, we also evaluated our method for DTW distance $D_{DTW}$ between GT and predicted strokes. To maintain the consistency of results for line-level and word-level methods, we evaluate all the methods for word-level metrics.

### 3.3 Results

The previous methods on IAM-online datasets work with line-level text for stroke trajectory recovery [3]. We initialize with the pre-trained model on the line-level text and finetune it for a word-level dataset with the proposed method.

Table 1 presents quantitative comparison, where bold numbers show the best results (lowest value). We observe that most metrics are much lower for word-level handwriting than the line-level input. These results show that separating the strokes from line-level text into words using flexible criteria for different handwriting improves the results compared to the line-level datasets. The results in the first and second rows of Table 1 show the evaluation for LSTM network trained with text-lines and words respectively. Note that, we report the results for network trained on text-lines (row 1) on word-level to keep the total number of strokes consistent with the methods trained on word-level.

Furthermore, the proposed addition of Gaussian weighting in the cost matrix in DTW loss (*word-level* $DTW_{G_{Align}}$) gives the lowest values for $\%N_{p,t}$ and $dist_{p,t}$, which mean that the predicted strokes are better imitating the GT strokes. We also validated our method for different values of variance ($\sigma = 2$ and $\sigma = 5$) in Gaussian function as shown in Figure 3. We do not go beyond $\sigma = 5$, since $\sigma = 5$ incurs sufficiently larger penalty for far-off points as shown in Figure 3. Table 1 shows the quantitative results for $\sigma = 2$ and $\sigma = 5$. Gaussian functions with

| Method | Line-level | word piece level | $DTW_{G_{Loss}}$ $sigma = 2$ | $DTW_{G_{Loss}}$ $sigma = 5$ | $DTW_{G_{Align}}$ $sigma = 2$ | $DTW_{G_{Align}}$ $sigma = 5$ |
|---|---|---|---|---|---|---|
| $Distance_{DTW}$ | 1.4058 | 1.1394 | 1.6908 | 2.8593 | 1.1258 | **1.0987** |

**Table 2.** DTW distance ($D_{DTW}$) between GT and predicted strokes

variance $\sigma = 2$ and $\sigma = 5$ have very close performance but $\sigma = 5$ have slightly better results.

The previous work [13] also uses Gaussian function in calculating the DTW loss, but our approach is methodologically different. [13] runs a regular DTW to align the stroke points and then computes the DTW loss weighted with Gaussian function. In contrast, we compute the stroke alignment using Gaussian weighting already in DTW table.

We tested the method in [13] for word-level strokes, and the results are listed in Table 1 under *word-level* $DTW_{G_{Loss}}$. The results for $DTW_{G_{Loss}}$ for $T_0 = 5$ are competitive with our method for $\%N_{t,p}$ and $dist_{t,p}$. However, our values for $\%N_{p,t}$ and $dist_{p,t}$ are better, which means that the predicted strokes are closer to GT strokes. In contrast, [13] can produce spurious predicted strokes, which is demonstrated in Figure 4.

This superiority of the proposed approach ($DTW_{G_{Align}}$) over [13] ($DTW_{G_{Loss}}$) is also validated by the DTW distances shown in Table 2, where our distances are significantly lower for both sigmas.

The visualization of recovered strokes in Figure 4 shows the superior quality of predicted strokes by our method ($DTW_{G_{Align}}$). This figure also shows that the proposed word level training (*word-level* $DTW_G$) gives better results than the line-level and word-level DTW. The red underline in Figure 4 shows incorrect recovered strokes. It can be seen that for line-level and word-level, there are missing or overlapping predicted strokes, wheres for $DTW_{G_{Loss}}$, there are spurious predicted strokes.

## 4 Conclusion

The proposed method for word-level stroke trajectory learning with Gaussian weighted cost matrix in DTW loss improves quantitative and qualitative results for handwriting stroke recovery.

## References

1. Abuhaiba, I.S., Holt, M.J., Datta, S.: Recognition of off-line cursive handwriting. Computer Vision and Image Understanding **71**(1), 19–38 (1998)
2. Agrawal, M., Bali, K., Madhvanath, S., Vuurpijl, L.: Upx: A new xml representation for annotated datasets of online handwriting data. In: Eighth International Conference on Document Analysis and Recognition (ICDAR'05). pp. 1161–1165. IEEE (2005)

3. Archibald, T., Poggemann, M., Chan, A., Martinez, T.: Trace: A differentiable approach to line-level stroke recovery for offline handwritten text. arXiv preprint arXiv:2105.11559 (2021)
4. Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: KDD workshop. vol. 10, pp. 359–370. Seattle, WA, USA: (1994)
5. Bhunia, A.K., Bhowmick, A., Bhunia, A.K., Konwer, A., Banerjee, P., Roy, P.P., Pal, U.: Handwriting trajectory recovery using end-to-end deep encoder-decoder network. In: 2018 24th International Conference on Pattern Recognition (ICPR). pp. 3639–3644. IEEE (2018)
6. Choi, W., Cho, J., Lee, S., Jung, Y.: Fast constrained dynamic time warping for similarity measure of time series data. IEEE Access **8**, 222841–222858 (2020)
7. Doermann, D.S., Rosenfeld, A.: Recovery of temporal information from static images of handwriting. International Journal of Computer Vision **15**(1-2), 143–164 (1995)
8. Jeong, Y.S., Jeong, M.K., Omitaomu, O.A.: Weighted dynamic time warping for time series classification. Pattern recognition **44**(9), 2231–2240 (2011)
9. Liu, C.L., Yin, F., Wang, D.H., Wang, Q.F.: Casia online and offline chinese handwriting databases. In: 2011 International Conference on Document Analysis and Recognition. pp. 37–41. IEEE (2011)
10. Marti, U.V., Bunke, H.: The iam-database: an english sentence database for offline handwriting recognition. International Journal on Document Analysis and Recognition **5**(1), 39–46 (2002)
11. Matsumoto, K., Fukushima, T., Nakagawa, M.: Collection and analysis of on-line handwritten japanese character patterns. In: Proceedings of Sixth International Conference on Document Analysis and Recognition. pp. 496–500. IEEE (2001)
12. Moussa, E., Lelore, T., Mouchère, H.: Applying end-to-end trainable approach on stroke extraction in handwritten math expressions images. In: ICDAR 2021: 16th International Conference on Document Analysis and Recognition (2021)
13. Nguyen, H.T., Nakamura, T., Nguyen, C.T., Nakawaga, M.: Online trajectory recovery from offline handwritten japanese kanji characters of multiple strokes. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 8320–8327. IEEE (2021)
14. Plamondon, R., Privitera, C.M.: The segmentation of cursive handwriting: an approach based on off-line recovery of the motor-temporal information. IEEE Transactions on Image Processing **8**(1), 80–91 (1999)
15. Plamondon, R., Srihari, S.N.: Online and off-line handwriting recognition: a comprehensive survey. IEEE Transactions on pattern analysis and machine intelligence **22**(1), 63–84 (2000)
16. Privitera, C.M., Plamondon, R.: A system for scanning and segmenting cursively handwritten words into basic strokes. In: Proceedings of 3rd International Conference on Document Analysis and Recognition. vol. 2, pp. 1047–1050. IEEE (1995)
17. Qiao, Y., Yasuhara, M.: Recover writing trajectory from multiple stroked image using bidirectional dynamic search. In: 18th International Conference on Pattern Recognition (ICPR'06). vol. 2, pp. 970–973. IEEE (2006)
18. Rabhi, B., Elbaati, A., Hamdi, Y., Alimi, A.M.: Handwriting recognition based on temporal order restored by the end-to-end system. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 1231–1236. IEEE (2019)
19. Viard-Gaudin, C., Lallican, P.M., Knerr, S.: Recognition-directed recovering of temporal information from handwriting images. Pattern Recognition Letters **26**(16), 2537–2548 (2005)

20. Viard-Gaudin, C., Lallican, P.M., Knerr, S.: Recognition-directed recovering of temporal information from handwriting images. Pattern Recognition Letters **26**(16), 2537–2548 (2005)


**Fig. 4.** Four examples to exhibit the visual quality of stroke recovery: In each example, first and second rows show the original handwriting and word division respectively, third and forth rows show stroke recovery from line-level and word level network, fifth and sixth rows show the strokes recovery by Gaussian weighted DTW and the proposed Gaussian weighted cost matrix with $\sigma = 5$. Each stroke is shown in different color (red, blue or green).