Slant Correction in an Optimization Framework

S. Uchida
Faculty of Information Science and Electrical Engineering
Kyushu University, Japan

In this presentation ....

Topic 1
Non-uniform slant correction for handwritten word

and

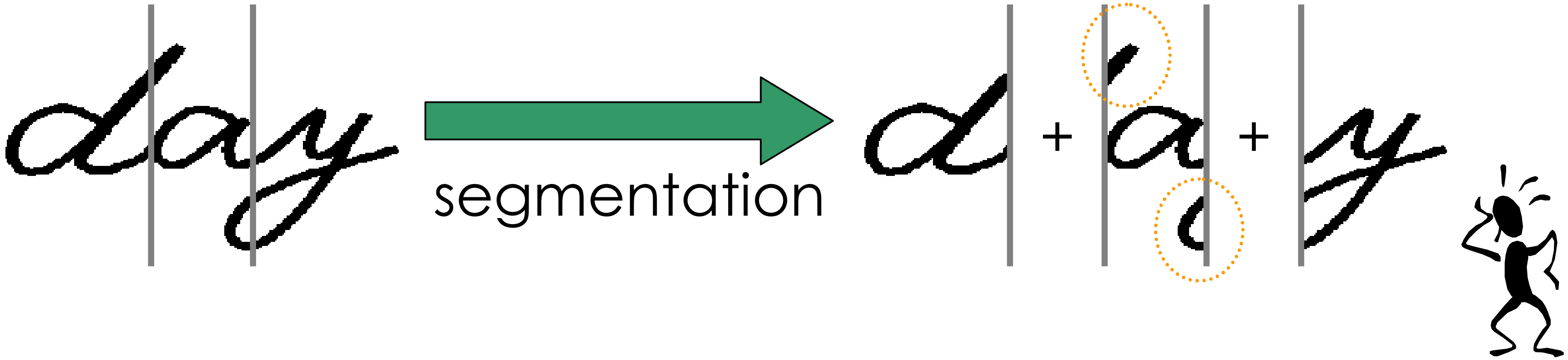
Topic 2 (planning phase)
Non-uniform skew correction for scanned document

Topic 1

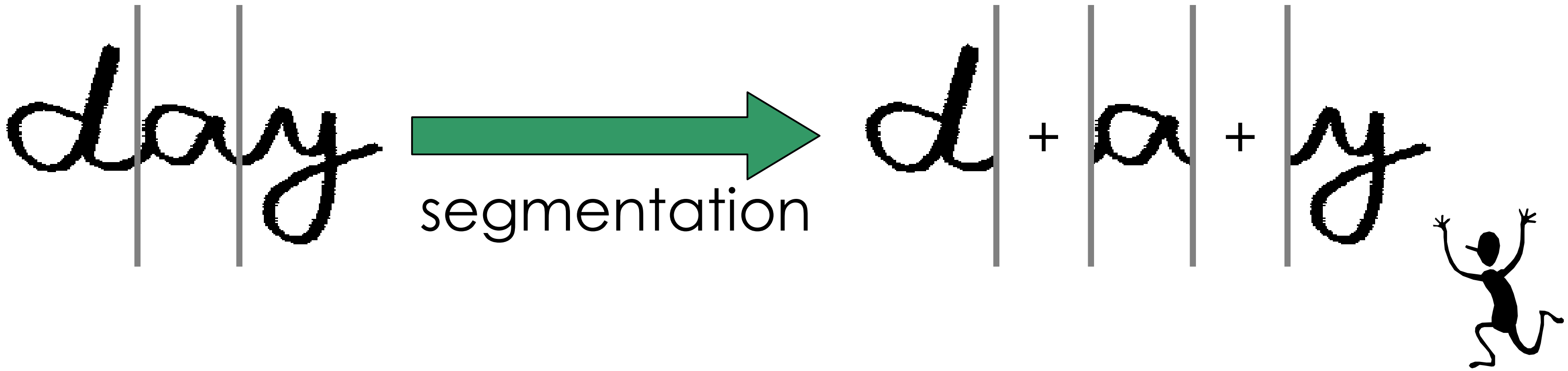
Non-uniform slant correction for handwritten word

Slant correction of handwritten word

Handwritten word

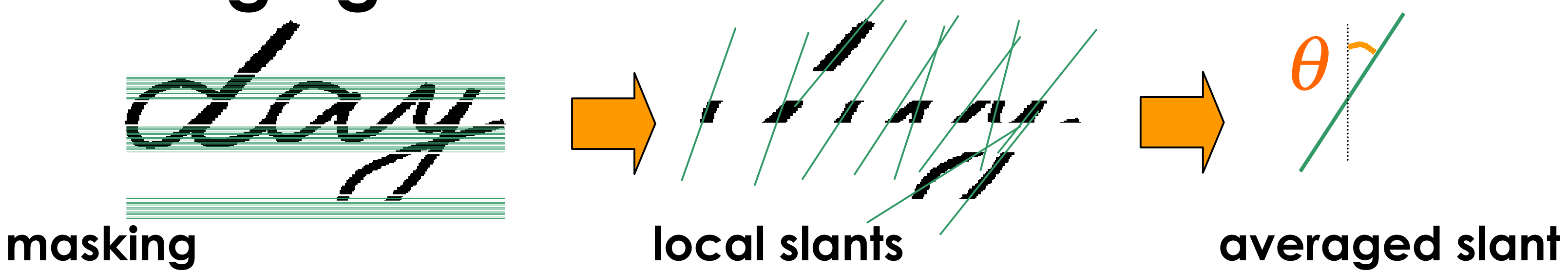


Slant-corrected word



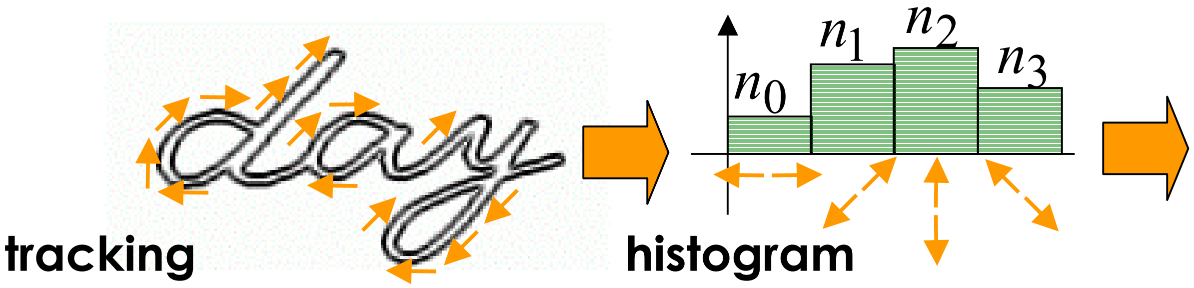
Conventional techniques

- Averaging



- Projection histograms

- Statistics of chain-coded stroke contours



$$\theta = \tan^{-1}\left(\frac{n_1 + n_2 + n_3}{n_1 - n_3}\right)$$

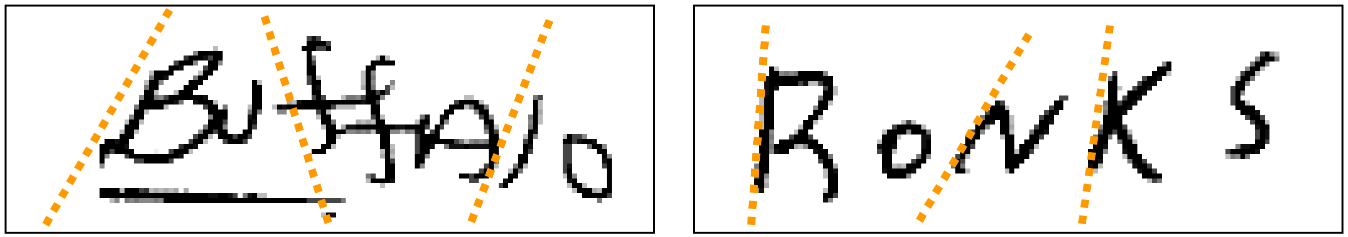
Conventional techniques : the problem

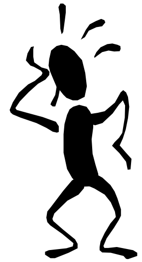
Assumption in all conventional techniques :

Slant is uniform in a word

But…. sometimes, slant is non-uniform

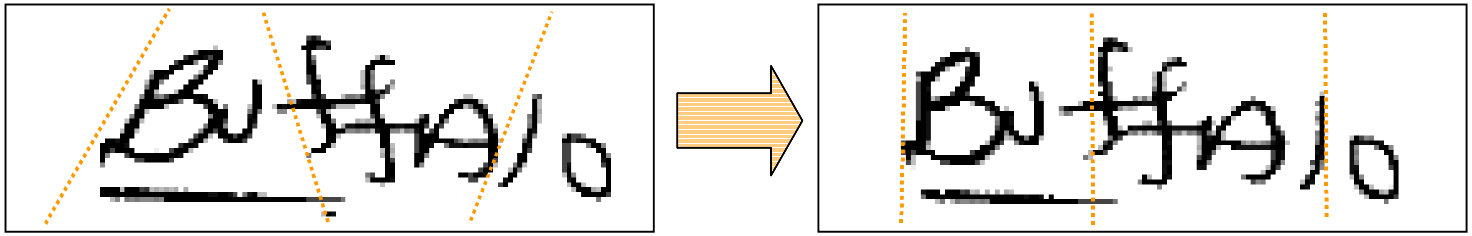
(with a probability of 5 to10%)

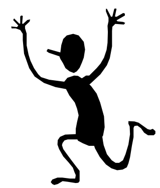




Overview of the present technique

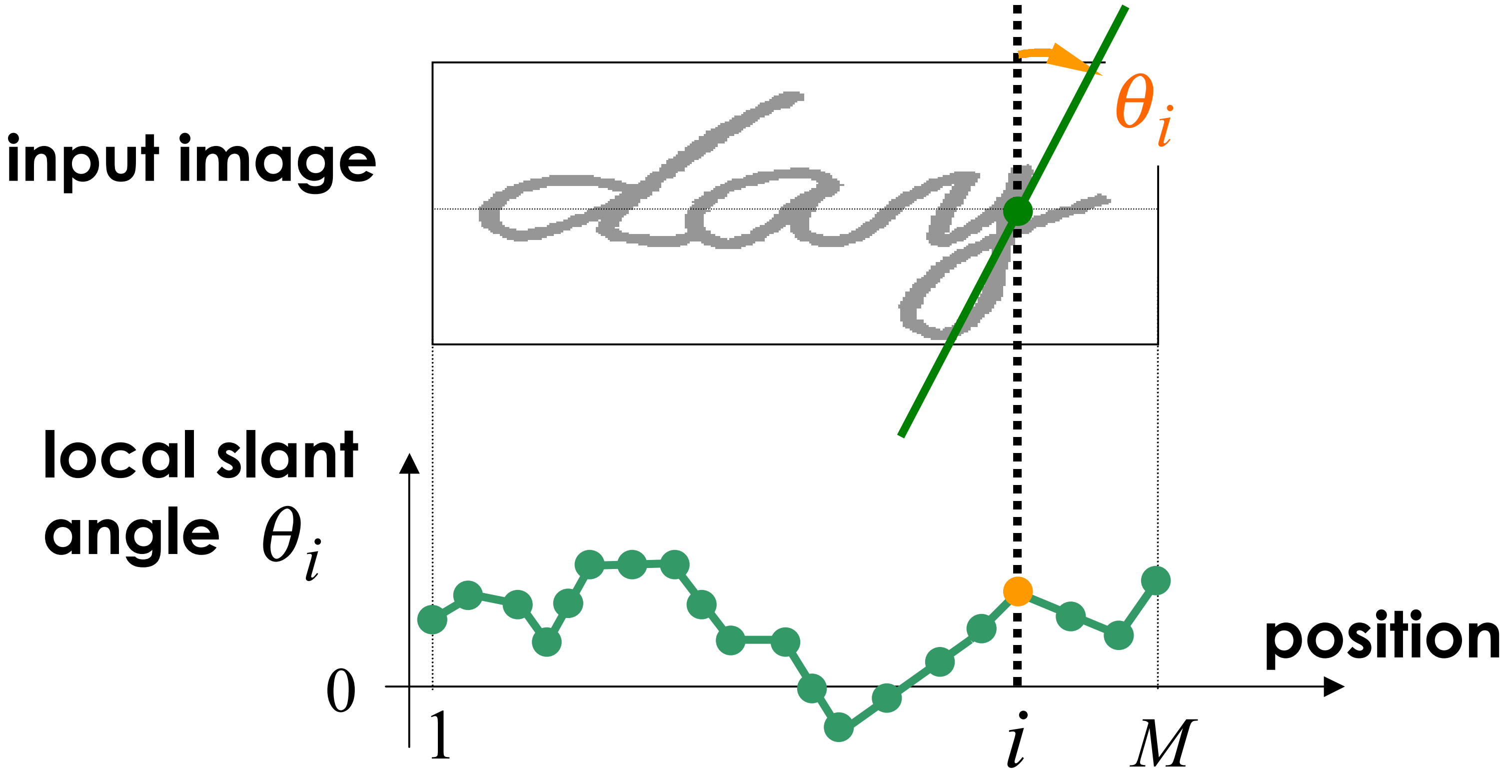
- Estimation and correction of non-uniform slant in an optimization framework





- Simple and fast algorithm based on dynamic programming (DP)

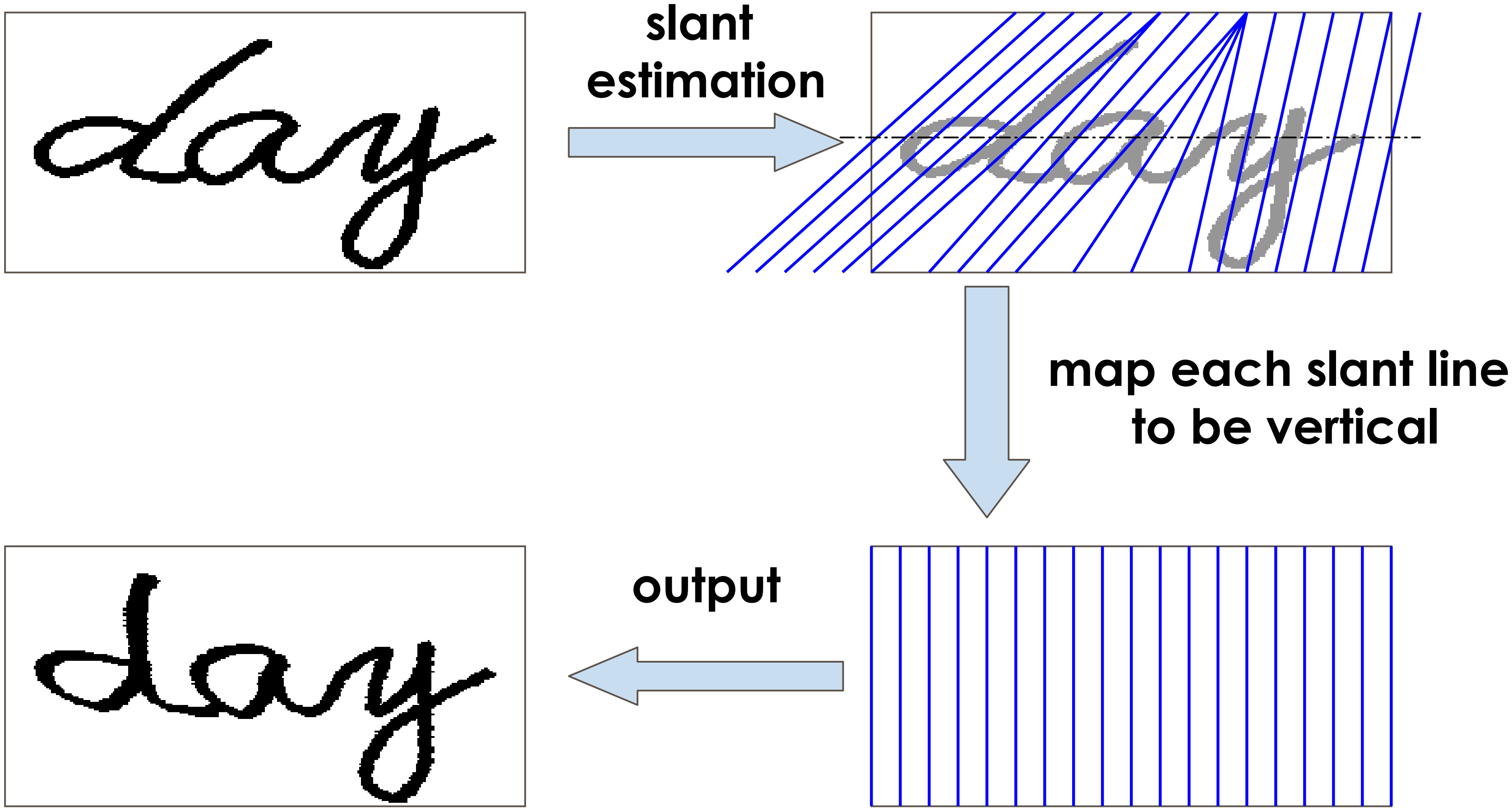
Non-uniform slant correction as an optimal estimation problem



non-uniform slant correction

= optimal estimation of $\theta_1, \ldots, \theta_i, \ldots \theta_M$

Create slant-corrected image using estimated slant



Directions on designing the criterion for the optimal estimation

Detect long vertical strokes

and

Propagate their slant angles to their neighborhood smoothly

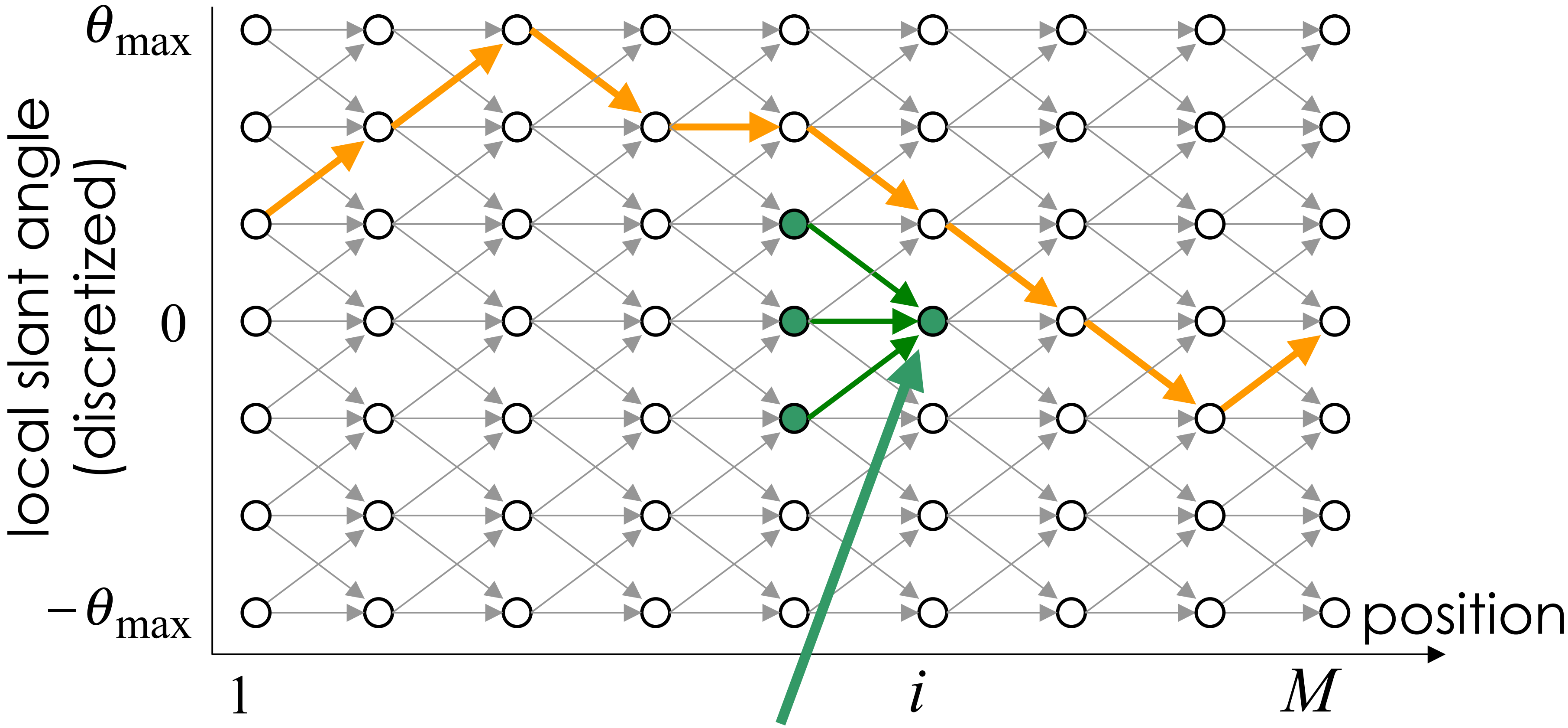
Our criterion

maximize $$\sum_{i=1}^{M}\left[s_i(\theta_i)+\rho(\theta_i \mid \theta_{i-1})\right]$$

- $s_i(\theta_i)$: confidence level of θ_i at position i

- $\rho(\theta_i \mid \theta_{i-1})$: continuity between θ_i and θ_{i-1}

Algorithm based on dynamic programming



DP recursion

$$g_i(\theta_i) = s_i(\theta_i) + \max_{\theta_{i-1}} \left[g_{i-1}(\theta_{i-1}) + \rho(\theta_i \mid \theta_{i-1}) \right]$$

Computational complexity

- **Theoretical :**

  $O(MNW)$

  where M=**width**, N=**height**, **and** W=**max slant**

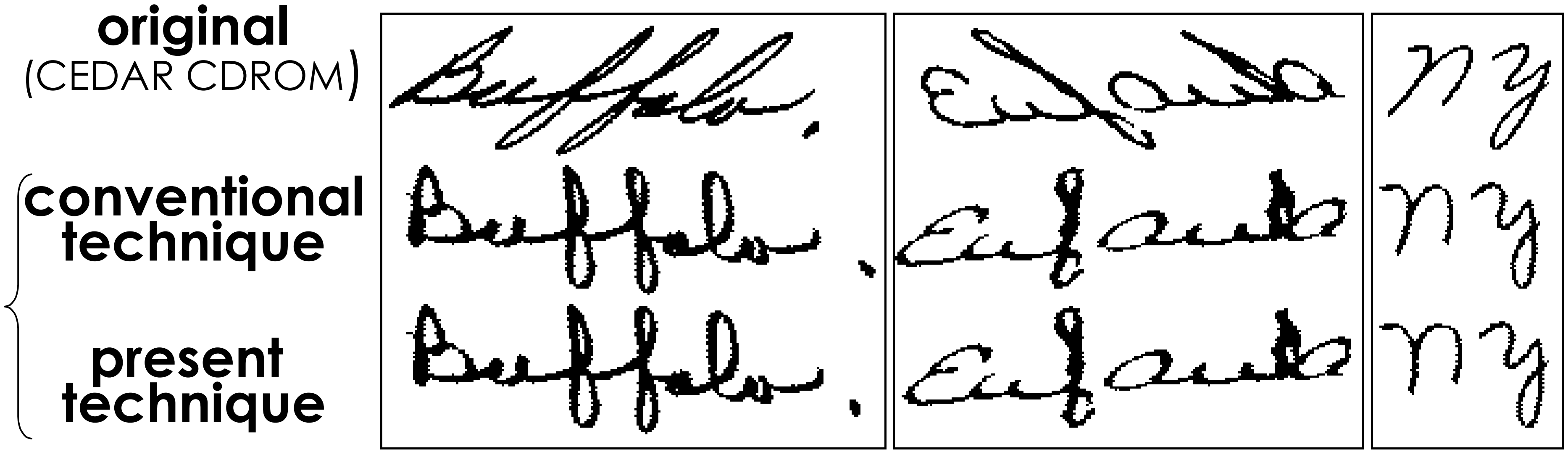
- **Practical :**

  140 ms (latest result)

  for M=256, N=64, **and** W=60(degrees)

  at PC with Pentium III, 500MHz

Results (1) : Correction of near uniform slant

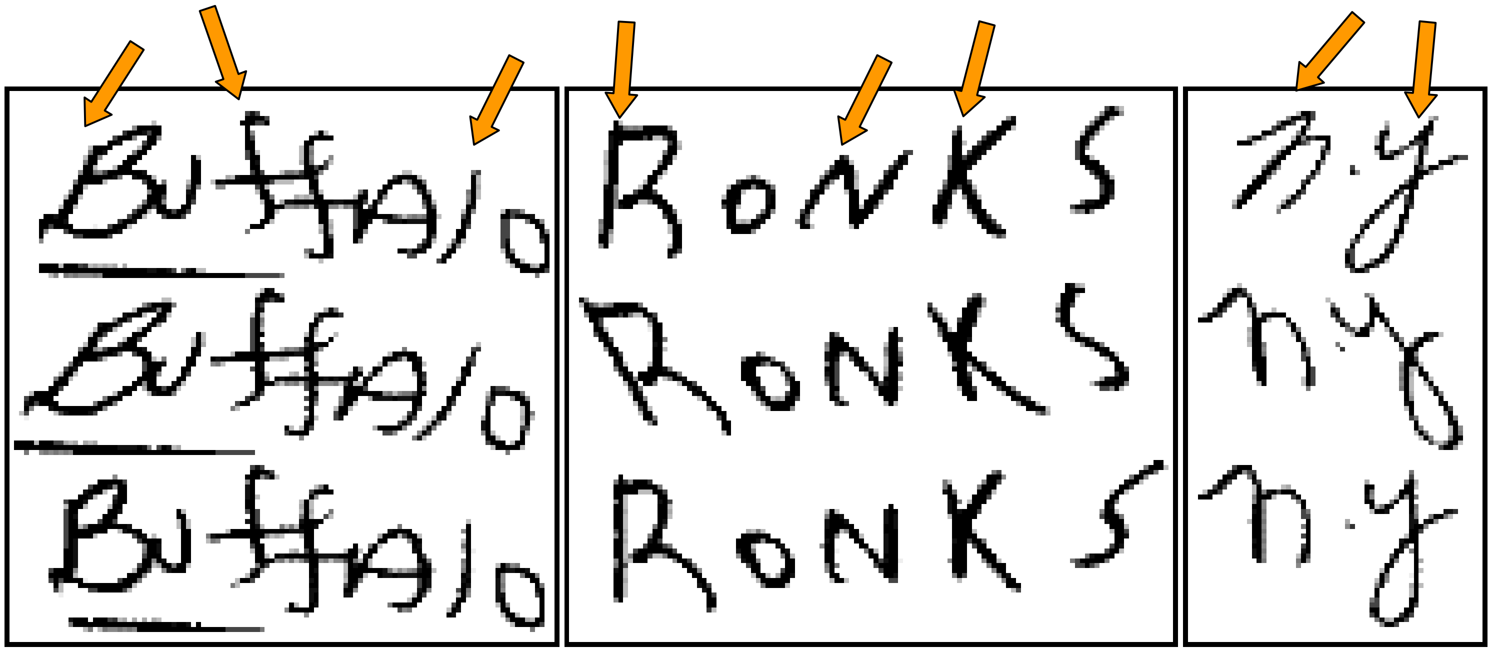


Results (2) : Correction of non-uniform slant

original
(CEDAR CDROM)

conventional technique

present technique

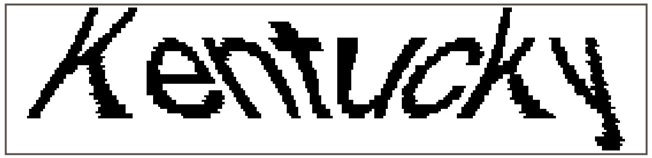


Quantitative evaluation (1)

- **Mean estimation error was measured using artificially slanted words**

- **Two slant patterns were used**

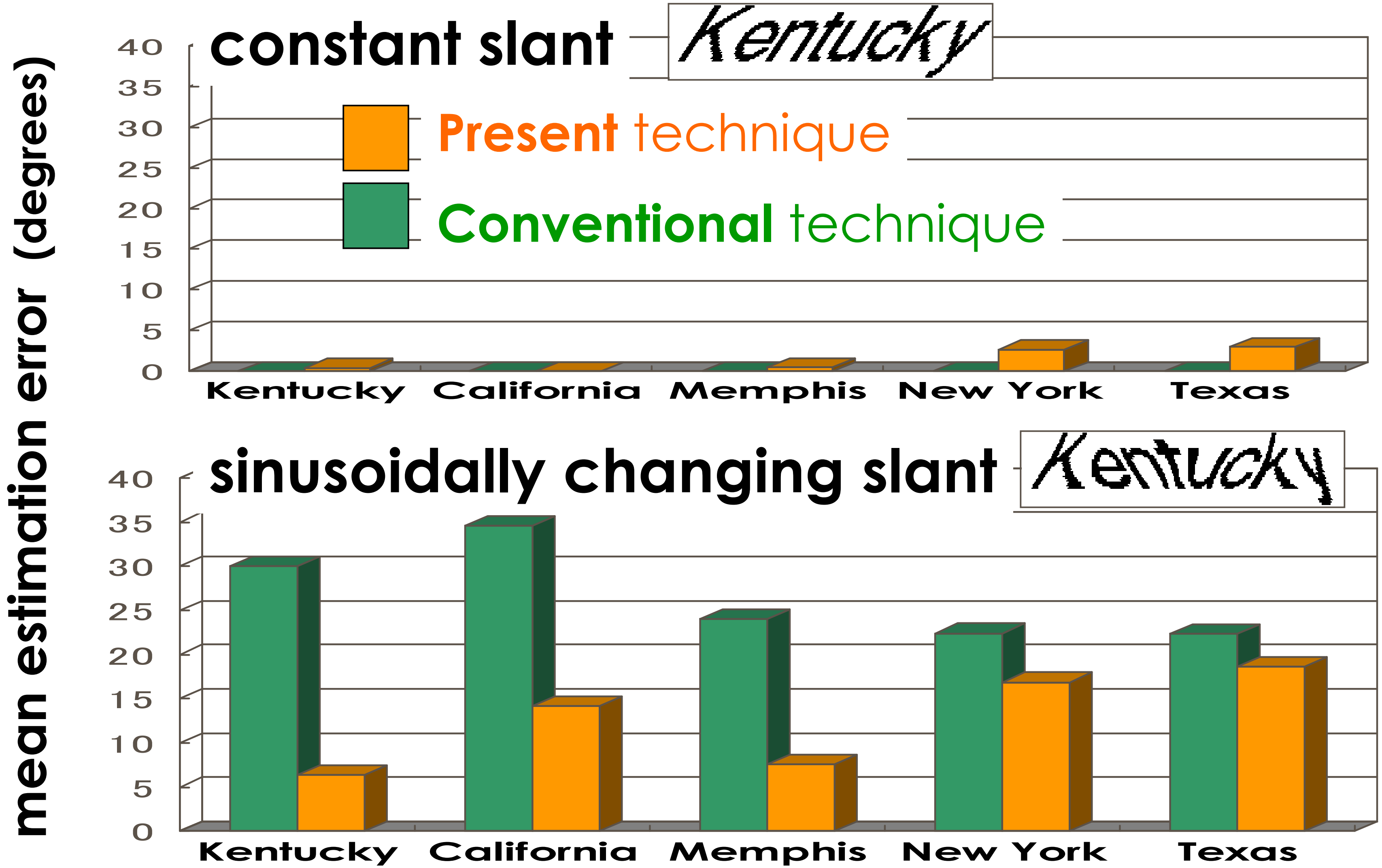
constant



sinusoidal

- **5 city name words were subjected**
  (Kentucky, California, Memphis, New York, Texas)

Quantitative evaluation (2)

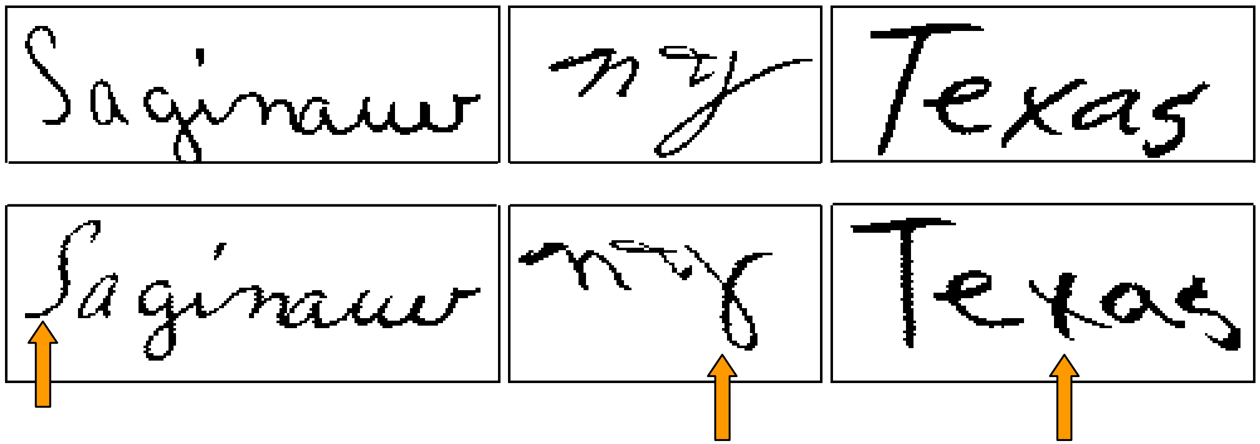


Remaining problem: over-correction

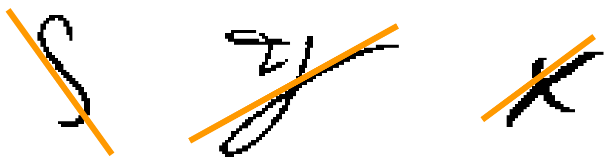
- Examples

original

present technique



- Cause : originally slanted parts



Conclusion of first part

- Formulate the non-uniform slant correction problem as an optimization problem
- Provide a simple and fast algorithm based on dynamic programming
- Indicate superiority over conventional techniques through experiments

Topic 2 (planning phase)

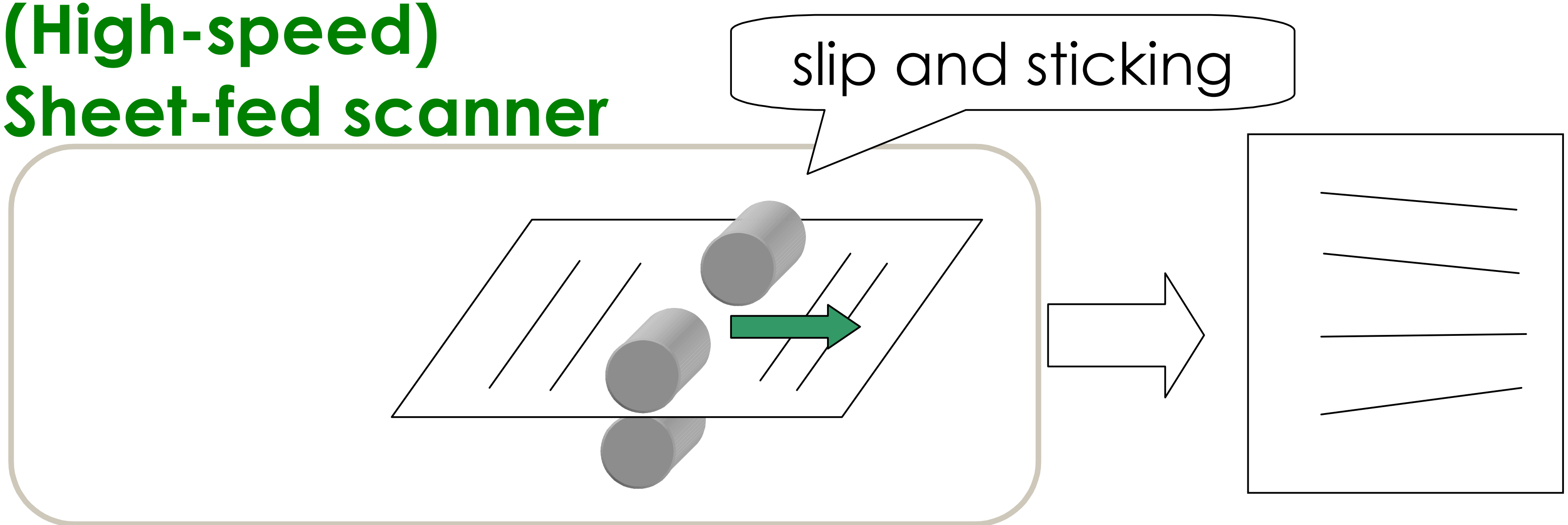
Non-uniform skew correction for scanned document

Purpose and methodology

- Remove non-uniform skew in scanned document
- Use an algorithm similar to the foregoing non-uniform slant correction algorithm

Types of non-uniform skew (1): Skew due to paper feeding

(High-speed) Sheet-fed scanner

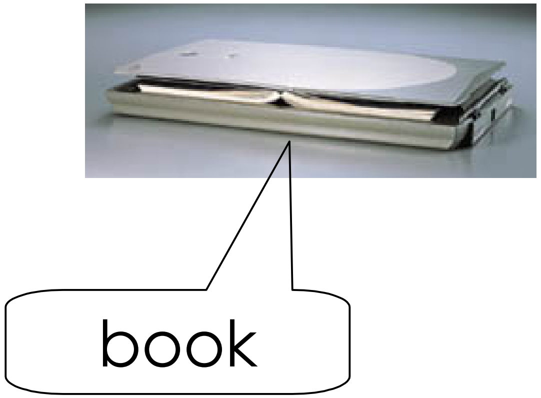


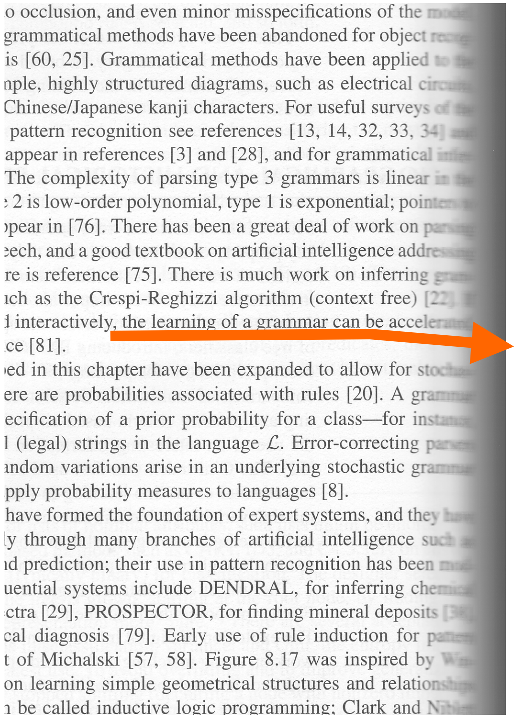
cf. Hirano-Okada-Yoda (2001, ICDAR)

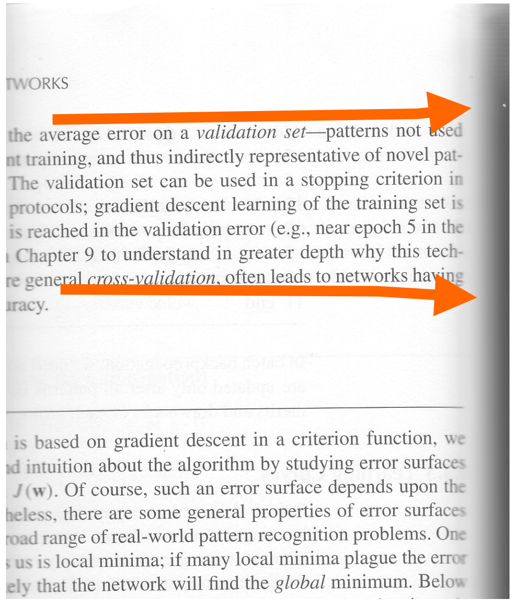
non-uniform skew correction by affine transformation

Types of non-uniform skew (2): Skew due to the thickness of book

Flatbed scanner




o occlusion, and even minor misspecifications of the
grammatical methods have been abandoned for object
is [60, 25]. Grammatical methods have been applied
nple, highly structured diagrams, such as electrical
Chinese/Japanese kanji characters. For useful surveys
pattern recognition see references [13, 14, 32, 33, 34]
appear in references [3] and [28], and for grammatical
The complexity of parsing type 3 grammars is linear
2 is low-order polynomial, type 1 is exponential;
ppear in [76]. There has been a great deal of work on
eech, and a good textbook on artificial intelligence
re is reference [75]. There is much work on inferring
ich as the Crespi-Reghizzi algorithm (context free) [22]
l interactively, the learning of a grammar can be
ce [81].
ed in this chapter have been expanded to allow for
ere are probabilities associated with rules [20]. A
ecification of a prior probability for a class—for
l (legal) strings in the language $\mathcal{L}$. Error-correcting
andom variations arise in an underlying stochastic
pply probability measures to languages [8].
have formed the foundation of expert systems, and they
y through many branches of artificial intelligence
d prediction; their use in pattern recognition has been
uential systems include DENDRAL, for inferring
ctra [29], PROSPECTOR, for finding mineral deposits
cal diagnosis [79]. Early use of rule induction for
t of Michalski [57, 58]. Figure 8.17 was inspired by
on learning simple geometrical structures and
n be called inductive logic programming; Clark and

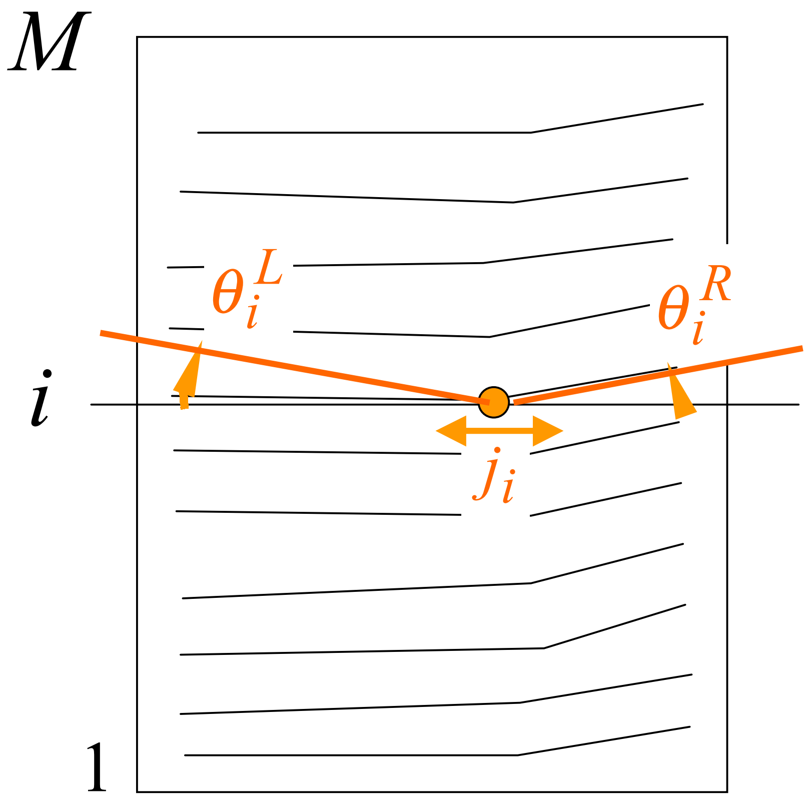

TWORKS

the average error on a *validation set*—patterns not used
nt training, and thus indirectly representative of novel pat-
The validation set can be used in a stopping criterion in
protocols; gradient descent learning of the training set is
is reached in the validation error (e.g., near epoch 5 in the
Chapter 9 to understand in greater depth why this tech-
re general *cross-validation*, often leads to networks having
uracy.

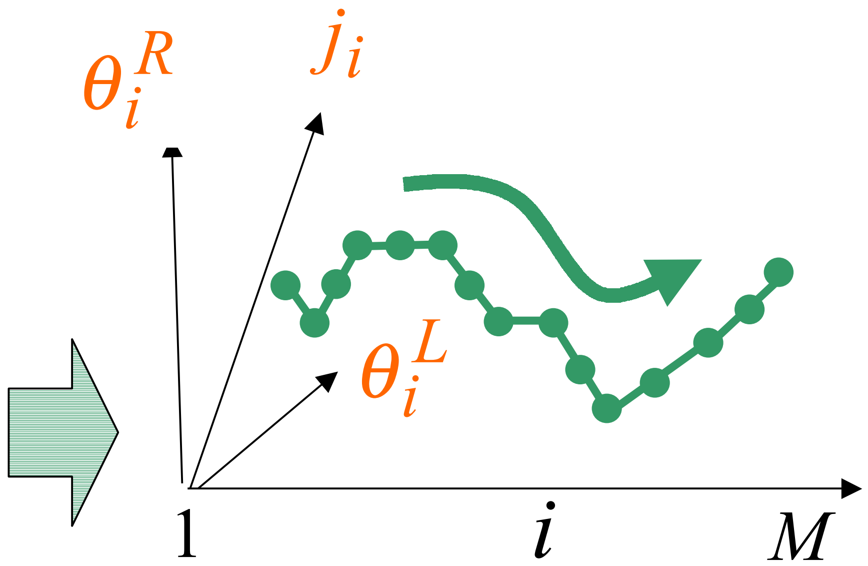
is based on gradient descent in a criterion function, we
d intuition about the algorithm by studying error surfaces
$J(\mathbf{w})$. Of course, such an error surface depends upon the
heless, there are some general properties of error surfaces
road range of real-world pattern recognition problems. One
us is local minima; if many local minima plague the error
ely that the network will find the *global* minimum. Below

Curvilinear distortion

Model of non-uniform skew

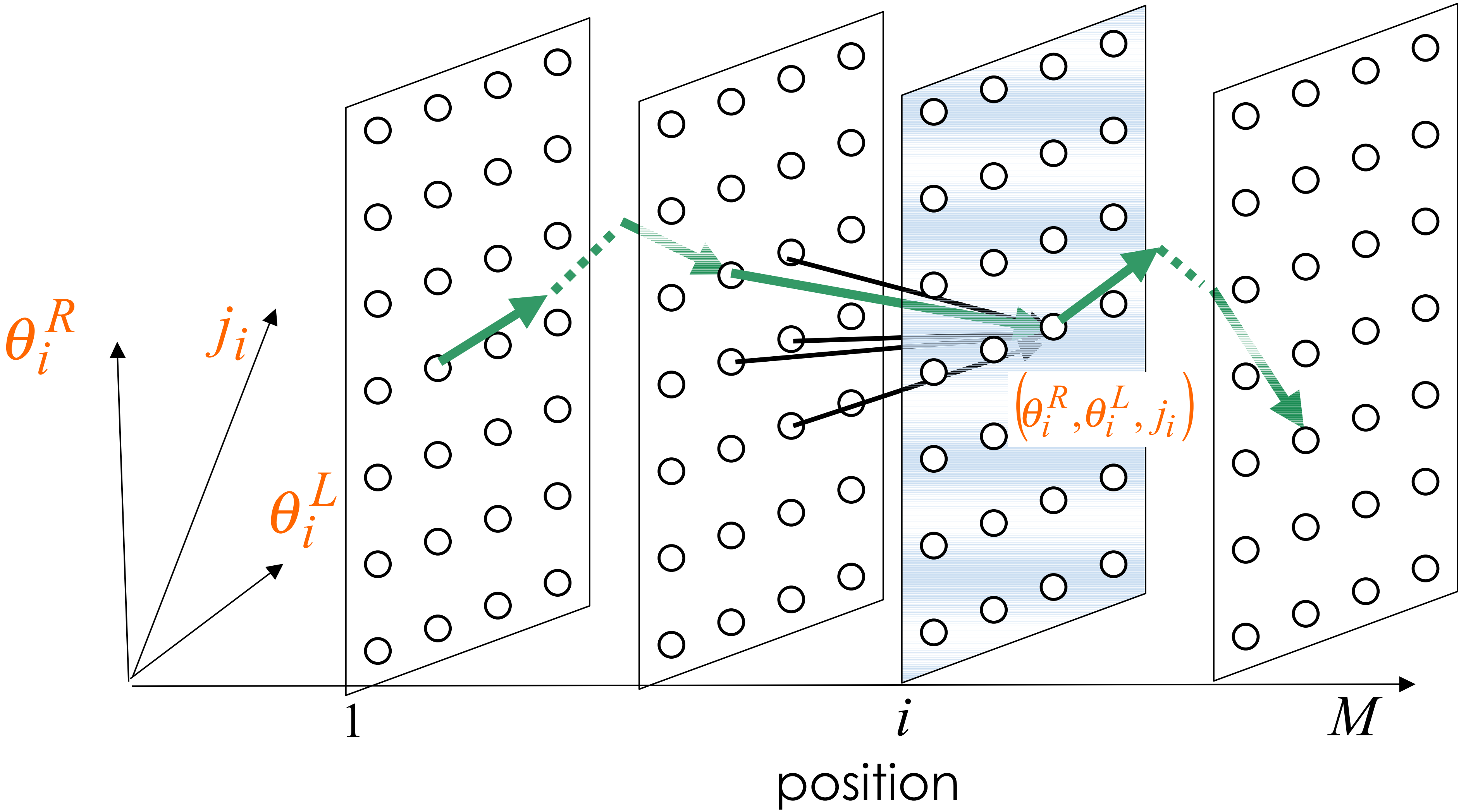


skewed document



non-uniform skew represented as the sequence of $(\theta_i^R, \theta_i^L, j_i)$

Algorithm based on dynamic programming



Future work for non-uniform skew correction

- Collect skewed documents and observe their characteristics
- Design criterion
- Evaluate the technique