Handwriting synthesis from handwritten glyphs

Isabelle Guyon

AT&T Bell Laboratories, Holmdel, NJ 07733
isabelle@research.att.com

Abstract

We present a very straightforward approach to the problem of handwriting synthesis. A writer provides once a set of handwritten samples of the most frequent groups of letters found in natural text (*glyphs*). Our program can then translate typed notes into natural looking handwritten notes from that writer. The method consists in assembling *glyphs* in a way that minimizes the total number of *glyphs* in each sentence.

1 Introduction

Considerable effort has been focusing in the past few years on the problem of handwriting recognition [1]. Pen computers have raised the hope that maybe handwriting recognition would substitute keyboards as a computer interface. Yet, many computer users find that keyboards are more efficient than handwriting and will not revert to using handwriting. Some users type almost everything they write because it is faster than handwriting. Others prefer typing because their handwriting is illegible.

In some cases, handwriting is preferable to typed text: it adds a personal touch. Most people handwrite their personal letters, greetings and compliments. Advertizers use handwriting to attract customers' attention.

This work was motivated by a remark of one of our executives who types virtually all his messages because typing is more expeditive. However, he would like sometimes to send personal handwritten note to his employees. Our method offers now such possibility.

The problem of handwriting synthesis is not new and many studies have been published in the literature. All the approaches that we are aware of make a real attempt at modeling the process of handwriting production [2, 3, 4]. Our approach is a lot more straightforward, but the result is surprising good and the transcription is very fast.

2 Description of the method

We extracted the most frequent groups of letters that appear in English text. This frequency analysis was performed on the Brown corpus [5], with the prefix tree method, previously described in Reference [6]. We added to this set all keyboard characters and two types of signature (first name only and formal signature). This constituted our *lexicon*, composed of approximately 1000 entries. The *lexicon* shown in Figure 1.

In our experiment, one writer (the author), entered one example of each entry from the list with a stylus on a digitizing tablet. We call *glyph* each handwriting sample. The tablet sampled the pen trajectory at a rate of approximately 100 points per second. It took approximately one hour to enter the entire *lexicon*.

We wrote a simple editor (see Figure 2) which allows the user to enter typed text and obtain a handwritten note with his own handwriting.

The handwritten text produced is the result of the juxtaposition of a series of *glyphs*. The sentence is first parsed into a sequence of groups of characters that belong to our *lexicon*, such as to minimize the number of cuts. In the example of Figure 2, the resulting parse is:

```
D|ear||A|li|ce|,||T|his||no|te||is||comp|ut|er||ge|ner|ated|.|
|C|an||you||be|li|eve||it|?||sign||
```

A cut is indicated by a vertical bar. A double vertical bar represents a space.

Our parser is written in the C programming language. The code was not optimized for speed, yet parsing such a short note in quasi instantaneous on a Sparc station 2. Most of the time to synthesize the handwritten note is taken by the graphics package that draws the glyphs. Parsing a full page of text takes a few seconds.

The result looks natural because of several reasons:

1. The same letter is never written is the same way in two different *glyphs*.
2. Ligatures within a *glyph* are natural.

The expert eye however cannot be fooled:

1. Some pen lifts are unexpected.
2. Some *glyphs* are repeated.
3. The inking is too regular.

It is only natural that we get some unexpected pen lifts since our method relies on groups of letters that are statistically frequent, as opposed to groups of letters that are naturally written together because of motor control constraints. Defining a *lexicon* based on motor control considerations would be more difficult and was left for further studies. Another way

sign (first name signature), Sign (formal signature), 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, /, ?, . (dot), ¿, ¡, ‘ (back quote), ~ (tilde), , (coma), ’ (quote), ” (double quote), ; (semi-colon), : (colon),], [, }, {, | — (vertical bar), =, +, _ (underscore), - (minus),), (, *, &, ^ (caret), %, $, #, , !, e, E, t, T, a, A, o, O, i, I, n, N, s, S, r, R, h, H, d, D, l, L, c, C, th, he, u, U, f, F, m, M, p, P, in, the, g, G, w, W, er, re, an, y, Y, on, b, B, ed, at, en, or, es, ti, v, V, to, te, nd, st, nt, ng, ar, it, of, ing, al, ha, as, ou, and, io, is, co, de, se, ve, ion, le, ai, ea, ro, me, id, k, K, ce, ent, li, ne, ll, sa, ri, ic, om, ra, fo, hi, il, tio, tion, si, pe, ec, pr, for, ta, la, be, ma, rs, di, ho, ut, aid, po, sai, said, tr, ca, ns, ur, ch, el, we, wa, wi, mi, ati, ts, ot, ee, un, no, rt, ac, ter, nc, ad, tha, ct, wh, ate, fi, ge, et, us, ol, hat, ie, her, pa, mo, ere, ers, ss, that, em, atio, ation, lo, na, ted, so, ly, con, ci, res, ni, os, ig, men, wo, ver, ow, ld, ir, ill, ul, su, vi, sh, ke, was, pl, pro, op, ment, ear, iv, ons, im, com, est, ov, ay, all, wit, fr, ther, with, ith, bu, ag, ia, bo, sp, mp, ev, his, are, nce, gh, av, rn, sta, fe, per, ep, ove, ab, am, x, X, ff, out, ort, gr, ry, tin, fro, bl, der, rom, by, oun, ive, do, tt, cl, ide, ef, from, rc, iti, tu, ex, go, ave, ect, ei, ty, tra, ga, cr, wer, ess, ls, int, red, eve, tor, rea, oul, ap, ting, pp, ba, cou, rat, j, J, over, nte, rm, cen, thi, cu, igh, not, str, up, ions, ye, uld, ist, man, oo, ore, ice, cti, ck, pre, lli, cent, ome, da, por, ies, hav, ial, ain, era, ine, one, fa, du, rd, par, uc, ey, have, din, art, oc, ern, mil, our, port, nde, ght, nts, eg, but, den, were, od, ding, has, off, sti, und, oth, if, end, au, ont, sed, inc, lea, ew, tat, fic, als, sin, nti, who, ast, led, age, rr, lin, ste, rce, ki, rin, af, use, min, ant, bi, illi, lat, ht, sc, ity, ight, eas, cal, rec, nat, rep, enc, ug, een, yea, year, ned, gi, abo, ud, whi, mm, here, pla, ak, whe, pu, ru, rou, cha, han, cia, omp, othe, sio, sion, lit, comp, ue, other, pi, nal, q, Q, act, ft, qu, hey, ead, ica, wou, rg, lio, ring, ffi, lion, wor, sid, wil, ose, hou, stat, ici, tur, ffic, ces, lle, lu, will, tes, ssi, ral, ele, mon, ure, dent, rk, ade, ctio, ction, cial, ds, tri, side, ded, pri, eme, eat, ntr, tic, bou, ree, ny, anc, eco, pres, ind, had, lic, dr, its, mor, part, um, new, erc, oli, any, unt, comm, nter, tw, ple, nati, offi, ase, sho, offic, sto, va, abou, tho, omm, bout, tim, nin, ins, pos, hin, tiv, pol, mb, thr, oi, rit, dis, ndi, own, inte, iden, cc, rie, ost, eo, ua, ble, nv, hr, tte, she, tate, bee, ub, fu, ona, ning, app, hei, been, spe, fir, cont, lan, mu, thei, kin, att, fte, hea, aft, this, ten, hic, nm, erce, ass, z, Z, poli, ren, emen, perc, ran, ated, ust, les, nn, ser, ack, ugh, wn, mpa, lt, oug, ough, ys, old, rati, tal, ju, ement, nme, nmen, rcent, rcen, ile, rl, ok, ake, br, ever, las, nment, xp, ili, afte, exp, vern, eral, orm, fter, ui, king, ire, nu, gov, gove, gover, owe, hen, ence, tive, eri, ount, ina, whic, more, vo, ote, esi, ident, onal, ini, ong, epo, ound, rv, pen, tl, eir, tro, ob, har, rte, iona, day, mill, ime, coun, ling, nst, sen, ous, gu, mar, yo, bil, wee, milli, gn, ord, ign, esti, form, cre, ance, epor, eport, ich, ven, nf, ner, ars, uri, som, lly, gre, repo, some, eop, isi, ves, sec, ope, erv, ard, air, che, fici, spo, two, ip, peo, ical, oll, ene, arg, gen, rge, ame, cte, nge, tar, sing, med, rnm, itio, sur, ople, opl, ition, duc, ita, thin, aw, fore, icia, count, enti, can, cted, ms, pec, acc, og, resi, inter, nes, ori, oin, eal, rate, iz, tie, say, cons, ks, sm, pt, chi, abl, ors, mer, ell, gra, last, eca, low, work, cau, ens, lar, nder, rad, ses, time, tan, iss, when, ib, than, uni, plan, bec, sse, you, arr, ffici, ang, oa, eli, pan, ied, hos, unde, cat, icial, gro, ric, ges, ary, rel, mat, rke, ompa, ace, sit, yi, ficia, late, ber, lie, vic, ters, nk, ecti, uti, ict, ties, omi, aus, ered, inv, ws, cit, mbe, ppo, esid, dec, ans, tel, ears, fer, eside, des, rest, ite, ese, ying, yin, edi, siden, now, ork, olic, ppe, cor, fl, hing, ture, serv, mber, los, ress, leg, nsi, ates, how, call, ilit, ret, lled, ause, arge, war, gai, gain, mit, nta, ann, lso, ivi, econ, hu, ond, hil, bli, ark, pea, rti, itt, inst, ecu, ffe, eek, vis, fou, arl, usi, eac, son, hel, ved, mis, fin, eci, ali, reat, alle, lati, ease, gs, bill, hre, clu, mpl, sts, even, car, ubl, ked, ris, tem, rme, vin, irs, rre, ail, rac, uct, sl, beca, dit, under, orte, ece, able, try, rst, sup, rem, acti, rov, jo, cur, mpan, lead, firs, ries, nis, mos, ays, ssion, ssio, ange, erat, week, nds, ced, ster, unc, eed, rvi, aga, stra, imp, win, tors, mark, ncl, tak, ders, cto, ctor, ach, tre, prov, ise, ains, rop, lec, irst, ani, ari, ged.

Figure 1: List of groups of characters that a user needs to handwrite once (the *lexicon*).

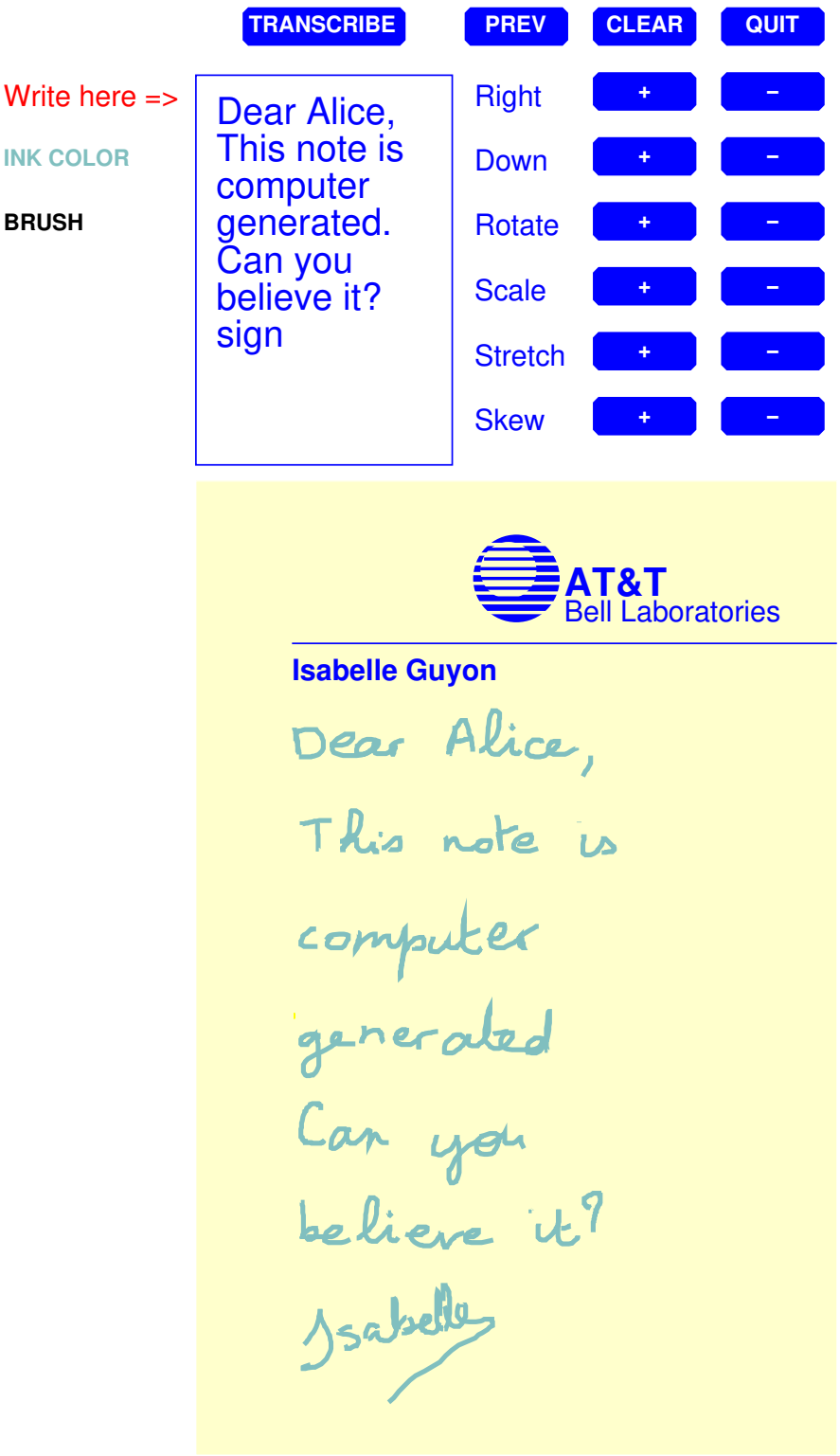

Figure 2: *Text-to-handwriting* editor.

of alleviating the problem of unexpected pen lifts would be to introduce a model of ligatures. We did not pursue this idea either.

We thought that the problem of *glyph* repetition would be obvious to the eye. However, in most examples, it requires a detailed examination of the handwritten note to discover a *glyph* repetition. Therefore, we concluded that one example of each *glyph* is enough. If we wanted to completely remove the problem of *glyph* repetition, rather that putting on the user the additional burden of writing several examples, we would introduce a distortion model to generate some shape variety. Several distortion models proposed in the literature would be suitable (see e.g. [7]).

We address the problem of inking in the next section.

3 Geometrical transformations and inking

Our simple editor provides a choice of ink colors and brush width. A variant of our example using a thin brush and dark ink is shown in Figure 3.

Inking with a brush that does not vary in thickness according to pen pressure variations betrays the synthesis artefact in an obvious way. The tablet that we used does not record pressure, so we could not restitute truly natural looking stokes. We made an attempt at introducing a more natural inking by using the pen trajectory direction and speed. The example of Figure 4 is simply obtained by enlarging the width of the line proportionally to the projection of the pen trajectory direction unit vector on a vertical axis, slightly tilted to the left. Using speed information did not improve the rendering.

Our editor also provides the possibility to move the text around, rotate it, scale it, stretch it and skew it. By combining such geometrical transformation, the handwriting style can be substantially modified. We show an example in Figure 5. Similar geometrical transformations can be used locally, instead of globally, to vary the shapes of identical glyphs that appear on the same page.

4 Other languages

Our *lexicon* was designed to reflect the statistics of English. Consequently, there is not guarantee that our model will work on other languages if we keep that particular *lexicon*. We tried to write a note in French (Figure 6) and the result came out quite all right, except for the accents. Therefore, it ought to be possible to obtain a language independent *lexicon* composed of group of frequent letters for the Latin alphabet, including all the accents of European languages. It is not expected that such a *lexicon* will have to be substantially larger that the present one. The same technique of handwriting synthesis could also be applied to other alphabets than the Latin alphabet.

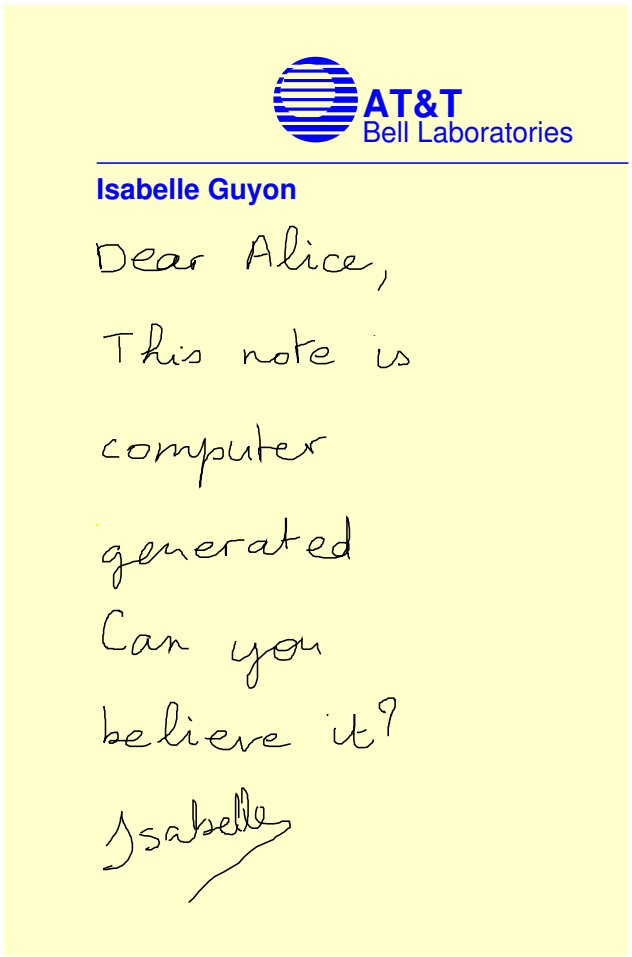

Figure 3: Example of Figure 2 with a thinner brush and darker ink.

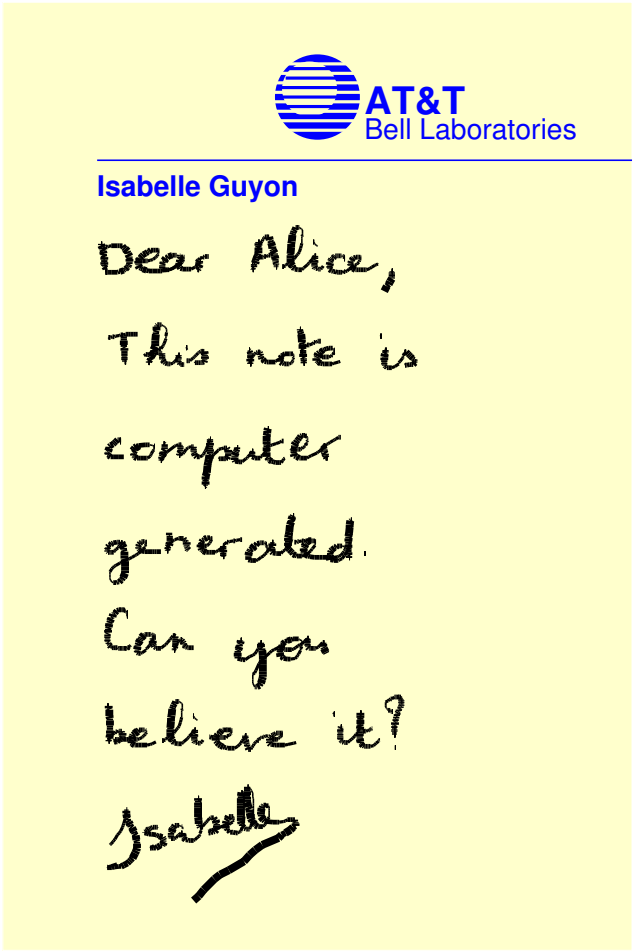

Figure 4: Example of Figure 2 with a fancier inking.

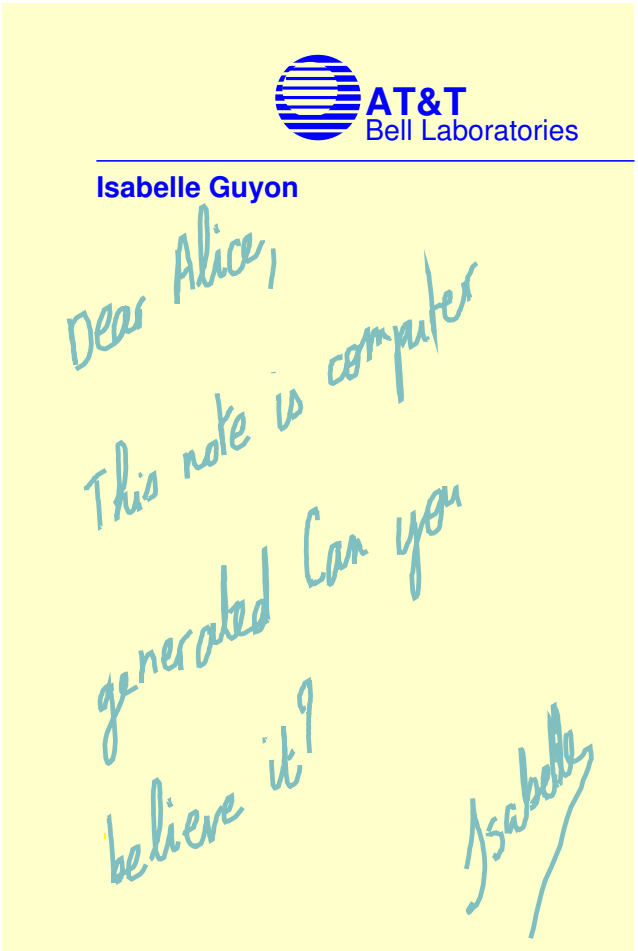

Figure 5: Example of Figure 2 after several geometrical transformation have been applied. Although the same *glyphs* are used, the handwriting style looks quite different.

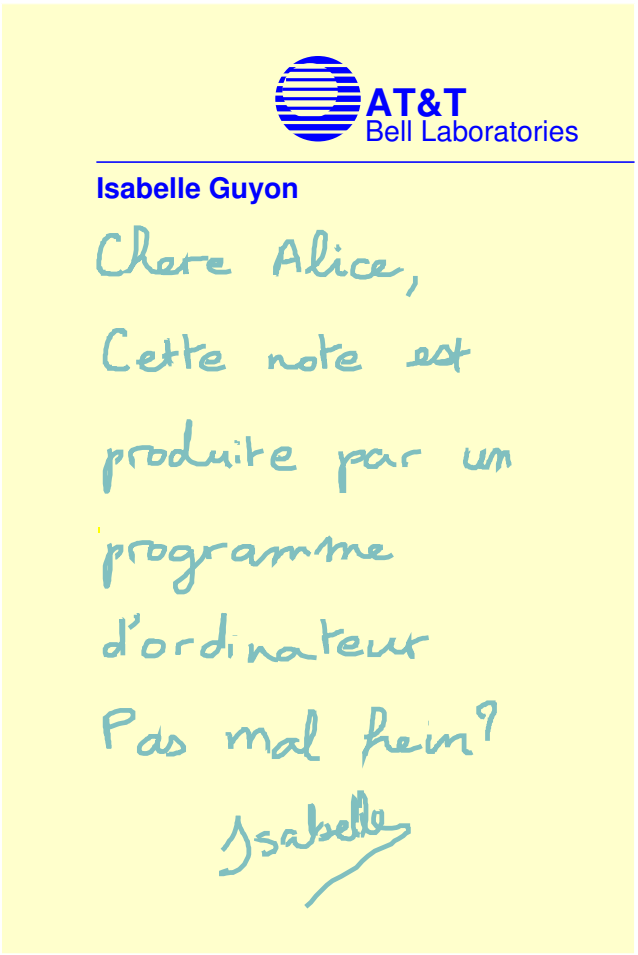

Figure 6: Example of Figure 2 translated into French (by the author).

5 Conclusion

We presented a method of handwriting synthesis whose simplicity is almost scary and yet produces natural looking handwriting. Improvements could be made by selecting glyphs with a motor control criterion (as opposed to a linguistic criterion), by varying glyph shapes with a distortion model and by inking proportionally to the pen pressure. We expect this simple and fast method of handwriting synthesis to find applications for writing personalized greeting adds and advertisements from typed text.

Acknowledgements

We would like to thank Arun Netravali for suggesting us to apply our previous work to handwriting synthesis.

References

[1] P. S. P. Wang and Bunke H., editors. *Handbook on Optical Character Recognition and Document Image Analysis*. World Scientific, 1996 (to appear).

[2] J.M. Hollerbach. An oscillation theory of handwriting. *Biological cybernetics*, pages 139–156, 1981.

[3] Singer Y. and Tishby N. Dynamical encoding of cursive handwriting. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1993.

[4] W. Guerfali and R. Plamondon. The Delta LogNormal theory for the generation and modelling of handwriting recognition. In *Proceedings of ICDAR'95*, volume I, pages 495–498. IEEE Computer Society Press, 1995.

[5] H. Kucera and W. Francis. *Computational analysis of present day American English.* Brown University Press, Providence, RI, 1967.

[6] I. Guyon and F. Pereira. Design of a linguistic postprocessor using variable memory length Markov models. In *Proceedings of ICDAR'95*, volume I, pages 454–457. IEEE Computer Society Press, 1995.

[7] P. Simard, Y. Le Cun, and J. Denker. Efficient pattern recognition using a new transformation distance. In S. Hanson et al, editor, *Advances in Neural Information Processing Systems 5*, pages 50–58, San Mateo CA, 1993. Morgan Kaufmann.