

3차원 방식의 수기 모사 연구

강태원*, 이기성*, 최원식⁺, 김태우**

(Manuscript received: Nov, 20, 2012 / Revised: Dec, 24, 2012 / Accepted: Dec, 28, 2012)

Study of 3D Simulator for Human Hand Writing

Tae-Won Kang*, Kee Sung Lee*, WonSik Choi⁺, Tae-Woo Kim**

Abstract

A light-weight enough to be portable yet functionally complete hand-writing machine was designed with hardware assembled and software implemented. The machine simulating human hand-writing not only allows identical massive scriptings on papers in human's proxy, but also offers emotional appearance as if the script was directly hand-written in ink. The features of human writing were independently put together to complete the action of scripting. The writing style could be altered in order to mimic a person's own specific writing style. The present work demonstrates that the assembled hand-writing machine was able to duplicate writing of scripts in almost parallel lines in consistent manner. The machine can be used in the emotional marketing area where seasonal greetings or thank-you cards are to be sent in massive manner. The pen-holding machine can also find an emotional application such as writing a short greeting note on a plastic lid on a disposable coffee cup in coffee-selling stores.

Key Words : Simulator(모사), Human hand writing(사람 글씨), 3D(3차원), Plotter(플로터), Code translation(코드변환)

1. 서론

현대 사회가 발전하면서 학문도 발전을 하고 이에 관련한 어플리케이션도 같이 발전하였으며, 그 중에서 과거에 비하여 급성장한 부분이 전자, 정보, 통신 분야라고 할 수 있다. 특히 컴퓨터의 발전은 글자 문화에 대한 크나큰 변혁을 이루어 왔다. 그 중에 하나가 컴퓨터 키보드 혹은 손글씨⁽¹⁾로 대표되는 글쓰기 문화에의 영향이다. 작가의 경우에도 예전에는 펜을 주로 사용하였으나 이제는 컴퓨터 앞에서 작업을 하는 것이 일반화되었다.

대개의 공학설계에서 장점만이 존재하거나 단점만이 존재하기보다는 장단점이 같이 공존하게 되는데, 컴퓨터를 이용한 글쓰기 또한 그러한 양면성을 가진다. 일례로 예전에는 초정장을 보내는 경우 또는 감사의 편지를 보내는 경우 친필을 사용하는 경우가 많았으나, 컴퓨터가 발전하면서 컴퓨터의 기능 가운데 하나인 워드 프로세서는 사람을 대신하여 깔끔한 글씨체로 정갈하게 문장을 만들어서 상대방에게 보내는 역할을 수행하게 되었다.

그러나, 디지털시대⁽²⁾에서 전통기술의 아날로그 시대를 그리워하듯, 감사의 편지 또는 중요한 고객에게 전하는 내용이 개인

* 국민대학교 기계시스템공학부

+ ㈜크레딕스

** 교신저자, 국민대학교 기계시스템공학부

주소: 서울 성북구 정릉3동

✉ Corresponding Author E-mail: twkim@kookmin.ac.kr

담당자가 친필로 써서 진심을 전하려는 마케팅이 선보이면서 인쇄된 필체를 사람이 대신하는 필요가 제기되었다. 그러나, 작가는 수백 장에서 많게는 수천 장 이상의 편지를 사람이 직접 작성하기에는 너무나 많은 에너지가 소요되어 시간과 노력 대비 경제적이지 않다.

하드웨어와 소프트웨어를 조합하여 다양한 목적을 구현하는 로봇⁽³⁻⁵⁾이나 시뮬레이터⁽⁶⁻⁹⁾에 대한 설계 및 제조연구가 수행되어 왔다. 그동안 로봇공학에서 기계를 대신하며 감성필기 목적을 동시에 구현하려는 일부 노력이 있었으나, 아직은 현실적인 대안이 나타나지 않은 상황이다.

따라서 본 연구는, 기존의 수기(hand writing) 기계가 디지털이제(digitizer)에서 읽은 값을 기준으로 글씨를 모방하는 경우에서 진일보하여, 올바른 글씨 인식 기술 및 응용성이 높은 개인 글씨체 생성 기술을 시현하려는 목표와 범위의 연구 결과이다.

2. 필기 모사 기술

2.1 필기체의 특성

개인들이 가지는 필기체의 특성을 결정하는 요소들을 살펴보면, 대략 6가지로 구분할 수 있다. 첫째는 글자의 크기이다. 글자의 크기는 쓰는 사람에 따라 그리고 음절 및 음소의 크기에 따라 균일하지 않다. 대부분은 초성의 크기가 다른 음소보다 크게 나타난다. 둘째는 획순이다. 정서일 때는 대부분의 사람들이 같은 획순을 가지지만 빠른 글자를 쓸 때는 그 순서가 다를 수도 있다. 특히 쌍자음일 때 그런 현상이 나타나곤 한다.

셋째, 흘림의 정도이다. 이는 주로 글자를 빨리 쓸 때와 천천히 쓸 때의 차이를 나타내는 요소이다. 넷째, 기울이기의 정도이다. 이 기울이는 X 축에 대해서 기울어질 때와 Y 축에 대해서 기울어질 때가 다른데, 대부분의 사람들은 Y 축을 기준으로 기울어지는 경향이 있다. 다섯째는 획수에 관한 것이다. 한 글자에 대해서 초성, 중성, 종성을 원래의 표준 획수대로 쓰는 경우는 없고, 사람마다 특징을 가지고 있다. 특히 초성과 중성을 붙여서 쓰는 경우 또는 중성과 종성을 붙여 쓰는 경우 일관성이 없다.

마지막으로 각 음소의 상대적인 위치 등이 필체⁽¹⁰⁻¹¹⁾를 결정하는 요소가 된다. 그러나 초, 중, 종성의 위치가 일정한 비율로 배치되지 않는 경우도 많다.

2.2 폰트의 특성

폰트 디자인에 사용되는 곡선은 다양하지만, 대개는 베이지어(Bezier), B-Spline과 그들의 여러 가지 변형 형태의 조합으로 이루어지는 것이 일반적이다. 그러나 베이지어 곡선은 제어점이 하나만 움직여도 모든 구역에서 그 곡선이 변하기 때문에 전체적인 윤곽을 추정하거나 부분적인 수정이 어렵다. 따라서

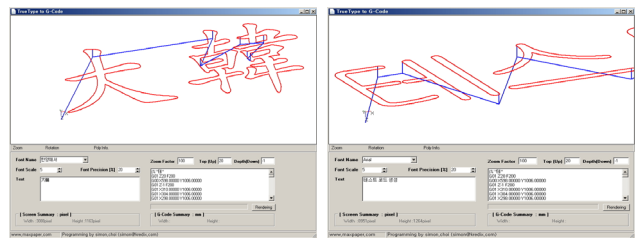


Fig. 1 TTF G-Code conversion in simulation screen

스플라인 곡선이 선호되고 있으며, 본 기술도 스플라인 곡선을 사용하였다. 특히, 스플라인 곡선의 경우 시작점과 끝점이 연결된다는 점, 시작점과 끝점을 동일한 점으로 간주하여 폐곡선이 가능하다는 점, 그리고 다각형으로 주어진 제어점에서 그 곡선의 추정이 가능하다는 점과 부분적인 수정이 가능하다는 점 등 여러 가지를 고려한 이유로 인하여 선택되었다.

2.3 Delta-Lognormal 이론

컴퓨터로 개인 폰트가 생성되는 과정도 중요하지만, 플로터(plotter)⁽¹²⁾ 또는 다른 출력 장치를 통하여 인쇄 하는 경우 사람이 직접 쓴 흔적 또는 결과를 얻기 위해서는, 펜을 사용하는 개인의 필기 정보가 어떻게 인식되는 가도 매우 중요하다. 이러한 측면에서 문제를 해결할 수 있는 이론이 필요하며, 필기생성 과정을 운동학적 모델을 근간으로 하는 delta-lognormal 이론이 가장 대표적이다. 이 이론은 펜 끝의 속도를 제어하는 함수들을 운동학에 기반하여 정의하고, 이의 파라미터 값을 조정함으로써 다양한 필기를 생성하도록 하는 방법의 근간이 된다.

한글 문자는 획의 연결이 이어지지 않는 경우가 많기 때문에 펜을 쓰다가 들어야만 한다. 그러나 빠르게 글씨를 쓰는 경우에는 이러한 불연속적인 펜의 움직임이 현저히 줄어든다. 따라서 글 쓰는 속도가 고려되어야 하고 펜올리기(pen-lifting)와 관련한 인자를 추출하여 펜을 들고 내리는 기구학적인 움직임에 대한 공식화(formulation)가 만들어진다.

본 연구에서는 이러한 운동학적인 파라미터를 이용하여 주어진 입력 데이터의 변형을 유도하는 것이 가능하다는 점을 사용하여 서체의 변형 또는 다양성이 적용되는 필기체의 생성이 가능하도록 모델링하는 것을 시도하였다.

Fig. 1은 본 기술을 적용하여 필기체를 생성하는 프로그램에 대한 예시이다.

3. 시제품 구성

사용자 편의성을 위하여, 실제 생성된 글꼴 데이터를 화면상에서 투영시켜주는 기능을 담당하게 된다. CNC 장비 특성상 많은 폴리곤을 빠른 속도로 처리해주어야 하며, XYZ축에 대한 확대, 축소 및 클리핑(clipping) 기능도 지원되어야 한다.

글꼴은 폴리곤으로 구성되며, 내부적으로는 직선과 곡선 (qubic-spline) 으로 구성되는데, 경로(routing path)엔진에서 최적화한 데이터를 사용하기 때문에, 직선으로 구성된다고 가정하여도 무방하게 된다.

또한, Z축 데이터가 존재하는데, Z축 데이터는 필기구의 접촉 유무를 결정하게 된다. 물리적인 장비의 특성상 모든 폴리곤은 연속으로 연결되어 있어야 하며, 사람이 글자나 자소 단위를 쓸 때, 글자간의 공백은 Z축을 올리거나 내리는 것으로 해결을 하게 된다. 이 부분이 일반적인 3D 플로팅(plotting) 엔진과는 다른 점이다.

Fig. 2와 같이 실제 필기모사 시스템에서는 물리적으로 XYZ 축이 연속적으로 움직이기 때문에, 연속적인 선분으로 표현해야 하며, 자소, 글자간의 이동은 Z축(파란색)이동으로 표시하게 된다.

또한 많은 글자를 화면에 고속으로 출력하기 위해서는 장비 종속적(device dependent)하게 출력할 수 있는 알고리즘이 필요하다. 이 알고리즘은 내부적으로 2개의 특성을 갖고 있다.

화면 해상도에 적합한 출력은 예를 들어 한 개의 글자가 약 100개의 선분으로 구성된다고 가정할 경우, Fig. 3에 보인 것처럼 “센터”는 200개의 선분으로 구성이 된다. 이 경우, 화면 해상도가 50x25픽셀일 경우에는 필요 없이 많은 겹쳐지는 선분이 존재하게 된다. 이럴 경우, 동적으로 동일한 선분을 제거를 하는 것이 바람직하여, 실제 글꼴의 좌표계를 사용하는 대신, 화면의 물리적 해상도 좌표계를 사용하여 계산하는 것이

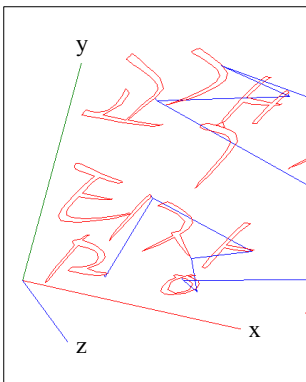


Fig. 2 Typical example of 3D plotting

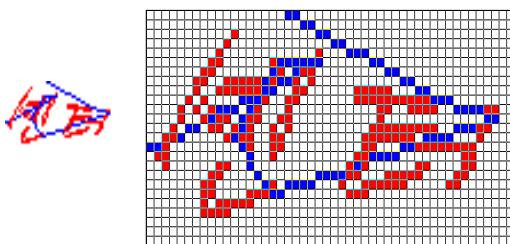


Fig. 3 Magnification (10 times) suitable for screen

훨씬 합리적이다. 본 시스템에서도 물리적 해상도를 사용하여 출력하므로, 화면 출력 속도의 개선을 갖게 된다.

Fig. 4의 영역과 같이 “센터”라는 부분만 화면에 보여져야 할 경우에는 전체 폴리곤 중에서 필요한 영역만 연산을 하여야 하며, 실제 클리핑 영역의 경우의 수는 총 4개의 경우의 수로 요약 가능하며, 교집합 영역을 계산하는 것으로, Fig. 5에 보인 것처럼 각각의 형태를 파악할 수 있다.

일반적으로 스텝핑 모터 제어를 위하여 사용하는 제어 방식은 펄스와 방향 두 가지 신호를 생성해서, 모터 드라이버에 제공하는 방식을 사용하게 되며, 마이크로스텝의 구현은 모터 드라이버 IC에서 처리하는 것이 시스템을 단순하게 만들 수 있는 장점이 있다.

본 시스템에서는 스텝모터의 회전수(rpm)의 탈조를 막기 위하여, 300으로 제한하였기 때문에, 마이크로 스텝을 사용하여 해상도를 높여야 한다.

또한 마이크로스텝을 이용하며, 가감속을 하기 위해서는 PC

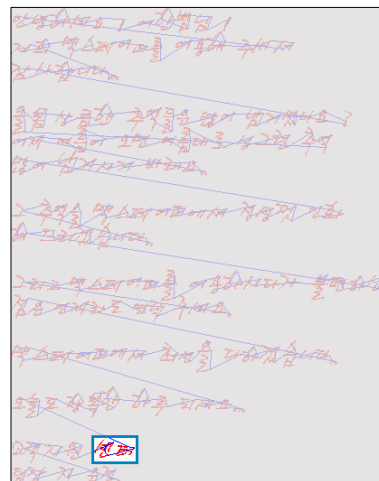


Fig. 4 Typical clipping for screen

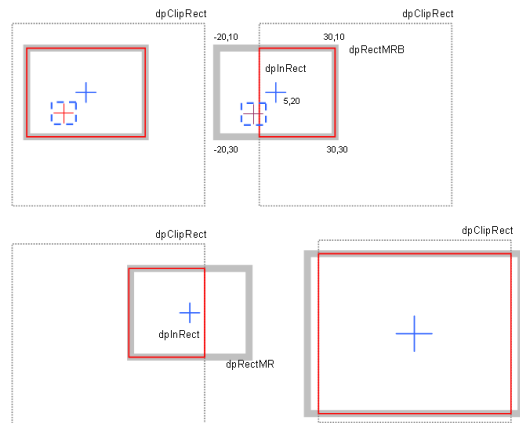


Fig. 5 Possible clipping regions

상에서 실시간 펄스 생성이 가능해야 하는데, 이는 Fig. 6과 같은 형태의 구성으로 자체 프로그램에서 16kHz이하의 신호에서는 약 1%이하의 정확도로 펄스 생성을 시킬 수 있다.

실시간 장비 조작을 위하여, 사용자가 수동으로 축 이동이 필요한 경우가 발생할 수 있으며, 이를 위해서는 Fig. 7에 보인 것처럼 직관적인 사용자 인터페이스와 운영알고리즘이 필요하다.

키보드를 누르는 동안에는 축 이동이 가능하며, Fig. 8과 같

```
// 실시간 Delay 함수
// CPU Clock
// 1.8Ghz -> 1,862,030,000
//           G M K
// Example
// 1 S      : 1862030000 Clk
// 1000ms   : 1862030000 CLK -> 1 mS : 1862030
// 1000000uSec : 1862030000 CLK -> 1 uS : 1862
// 1000000uSec : 1862030000 CLK -> X uS : ? Cnt
// X = 1862030000 * x / 1000000
//
// 예제
// 1.8각도
// 1회전 : (360/1.8)/2 = 400 pulse * 10 회전 = 4K Hz
//           20 회전 = 8K Hz
//
// Duty 50% 가정
//-----
// 2KHz : 1000000uS / 4000Khz = 250.0 [uS]
// 4KHz : 1000000uS / 8000Khz = 125.0 [uS]
// 8Khz : 1000000uS / 16000Khz = 62.5 [uS]
// 16Khz : 1000000uS / 32000Khz = 31.25 [uS]
// 32Khz : 1000000uS / 64000Khz = 15.62 [uS]
//
Procedure CNC_uDelay(uDelay : Int64);
Var
hrRes, hrT1, hrT2, dif: Int64;
begin
if QueryPerformanceFrequency(hrRes) then
begin
dif := Round(hrRes * uDelay / 1000000);
QueryPerformanceCounter(hrT1);
repeat
QueryPerformanceCounter(hrT2);
until (hrT2-hrT1) > dif;
end;
end;
end;
```

Fig. 6 Routine realizing real-time pulse generation

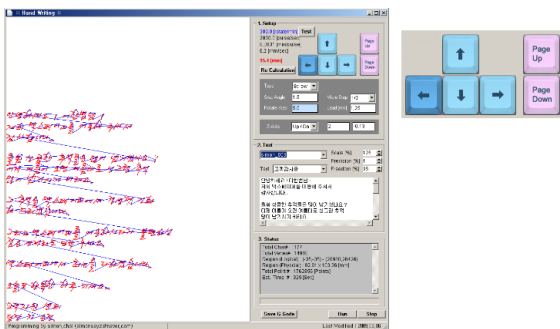


Fig. 7 Interface handling 3-axis operation

은 스테이트 머신을 통해서, 실제 운영 시에도 구현이 가능하다. 본 시스템에서는 두 개의 작업(thread)이 아래 방식과 같이 동시에 실행되게 된다.

가상 기계(virtual machine)를 통하여 실제 장비 없이도 결과치와 정확도를 예상할 수 있어야 하며, 이는 시뮬레이터를 통하여 가능해진다.

시뮬레이터의 정확도를 높이기 위해서는, Fig. 9(a)에 보인 것처럼 실제 장비 제어 루틴과 동일한 코드 (단일 코드)를 사용

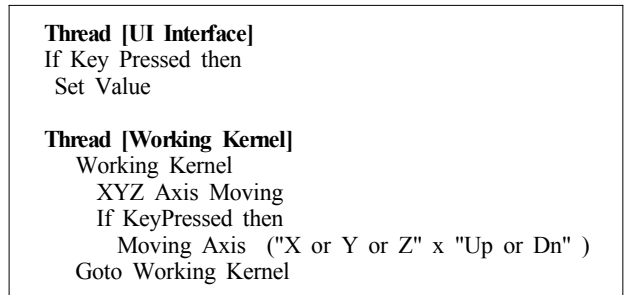


Fig. 8 Typical example for thread job

Case 1. Bolt Screw(lead 1.25 mm), 1.8-degree, microstep 1/2

Total Char# : 177
 Total Vertex# : 14960
 Region (Logical) : (-34,-34) - (21001,26309)
 Region (Physical) : 82.17 × 102.90[mm]
 Total Point # : 1766611 [Points]
 Est. Time # : 927 [Sec]

(a)

300.0 [rotate/min]	: Motor RPM
2000.0 [pulse/sec]	: Motor signal frequency
0.0031 [mm/pulse]	: equipment resolution
8.2 [mm/sec]	: equipment speed (1 axis)
15.5 [min]	: actual printing time
Total Char#	: 177, total printing characters
Total Vertex#	: 14960, total Vertex
Region (Logical)	: (-34, -34) - (21001, 26309) coordinate
Region (Physical)	: 82.17 × 102.90 [mm] real area for the equipment
Total Point #	: 1766611 [Points] Total points created for Vertex
Est. Time #	: 927 [Sec] Estimated time

(b)

Fig. 9 (a) System simulator, (b) meaning of results

Table 1 Tabulated results for simulation

rev/sec	angle	performance capability	lead	speed	Hz	print area	resolution	time (minute)
5	1.8	1/2	1.25	6.2	2K	64×82	0.0031	15.5
5	1.8	1/16	5	25	16K	32×41	0.0016	1.6
5	1.8	1/8	5	25	8K	67×82	0.0031	3.1
5	1.8	1/4	5	25	4K	131×165	0.0062	6.3
5	1.8	1/2	5	25	2K	260×329	0.0125	12.5
5	1.8	1/16	10	50	16K	66×82	0.0031	1.6
5	1.8	1/8	10	50	8K	132×164	0.0062	3.1
5	1.8	1/4	10	50	4K	268×328	0.0062	6.3
5	1.8	1/2	10	50	2K	507×657	0.0250	12.5
5	1.8	1/16	20	100	16K	127×165	0.0016	1.6
5	1.8	1/8	20	100	8K	263×329	0.0125	3.1
5	1.8	1/4	20	100	4K	530×657	0.0062	6.3
5	1.8	1/2	20	100	2K	1049×1318	0.0500	12.5

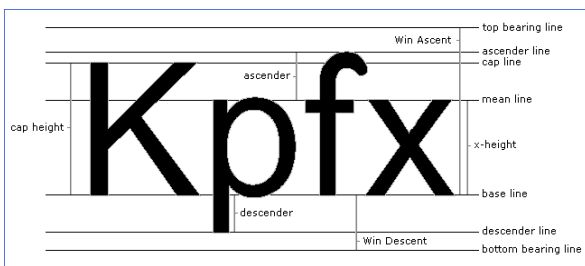


Fig. 10 Basic component for font change system

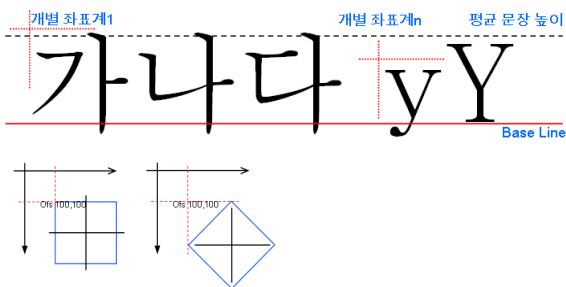


Fig. 11 Coordinate system for font changing system

하여 정확도를 검사하게 되며, 실제 장비와 약 5%정도 오차 범위안의 결과를 얻을 수 있음이 확인되었다. Fig 9(a)에서의 시뮬레이션 결과 값은 Fig 9(b)의 의미를 갖는다.

실제 생성된 글꼴에서 특성꼭지점(vertex)가 생성되면, 이 특성꼭지점을 기준으로 Table 1에 보인 바와 같이 실제 물리적 이송 거리를 선형 보간에 의해서 구하며, 이 보간된 전체 이동 포인트가 총 (total point) 수가 된다.

글꼴은 몇 가지 방식으로 변형될 수 있으며, 실제 구현 시에는 속도 문제로 인하여 간단한 변형만 1차 구현하였다. 좌표계

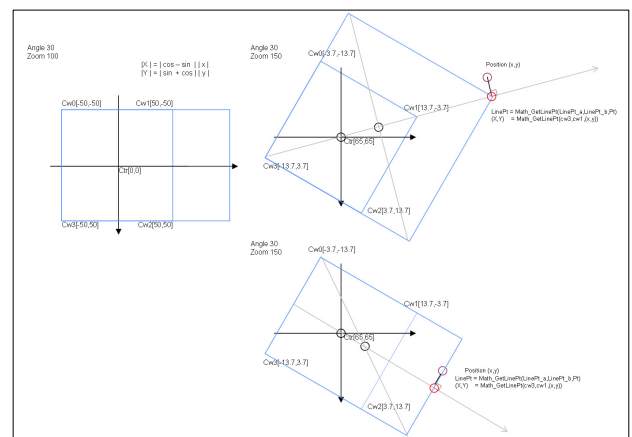


Fig. 12 Magnification or reduction in rotational direction

는 직각 좌표계를 사용하였으며, 베이스 라인에서 높이와 길이를 추출하였다(Fig. 10참조).

전체 문장에서 라인(줄)단위로 분석을 한 후에, 글자를 개별 좌표계로 분리, 변형을 시킨 후, 다시 재조합을 하는 과정을 거치게 된다.

전체 문장에서 라인(줄)단위로 분석을 한 후에, 글자를 개별 좌표계로 분리, 변형을 시킨 후, 다시 재조합을 하는 과정을 거치게 된다(Fig. 11참조).

Fig. 11과 같이 글꼴은 다시 자소단위로 분리시킬 수 있기 때문에, 회귀적으로 변형을 시키는 과정을 거칠 수도 있다.

문자의 재조합에서는 앞에서 설명한 방식으로 계산을 할 경우, 변형 오류가 발생할 수 때문에, 아래 그림과 같이 역계산후 재계산의 2번의 단계를 거치게 된다(Fig. 12참조).

이와 같은 형태의 변형을 반복시켜서 Fig. 13과 같은 결과를

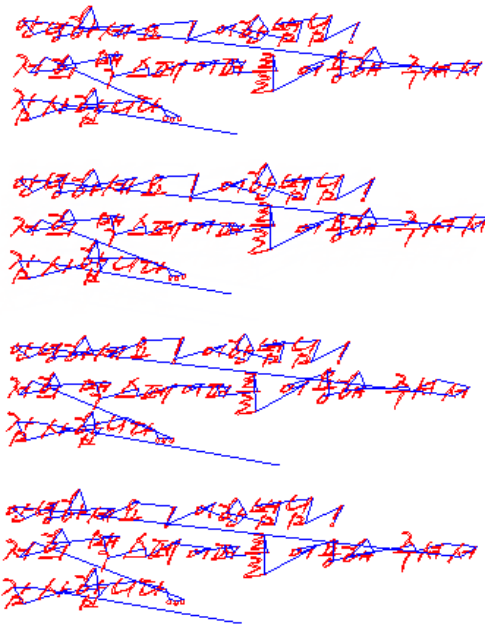


Fig. 13 Final computer writing simulation results

언을 수 있게 된다.

동일한 필체라고 인정할 수 있는 범위 안에서 자간, 장평, 문장의 길이(width), 높이(height)가 모두 틀리게 변형된 결과를 얻게 된다.

4. 모사적용 및 평가

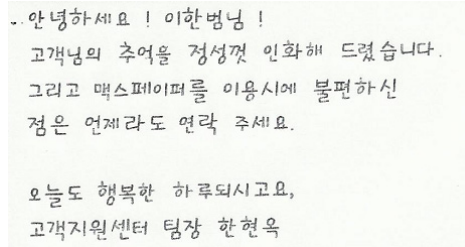
개발된 모사 시스템의 성능을 파악하고자 실제로 사람이 직접 작성한 글씨와 컴퓨터가 인식한 사람의 폰트를 기반으로 하여 다양한 필기체를 모사한 결과를 비교하였다(Fig. 14 참조). 실제 글씨를 모사하여 하드웨어적으로 볼펜이 쓴 글씨를 보인 Fig. 14 (b), (c), (d)는 사람의 글씨를 조금 과장되게 표현한 것이다. 즉, Table 1에서 보여주는 모사 인자들의 값을 변형한 결과들이다.

또한 Fig. 14 (d)는 한번 사람 글씨에 대한 맵시가 결정이 되면 그 편지 내용은 컴퓨터에서 얼마든지 수정이 가능하여 다양한 편지내용과 콘텐츠를 만들어 낼 수 있음을 보여주는 모사기능의 확장 예이다.

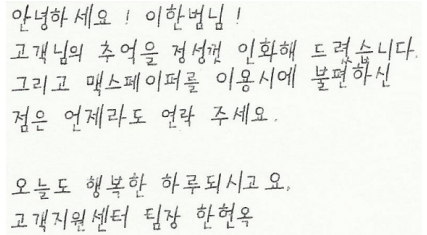
5. 결론

사람의 글씨를 흉내낼 수 있는 모사기계 시스템을 구축하였으며, 테스트를 해본결과, 다음과 같은 결론을 얻었다.

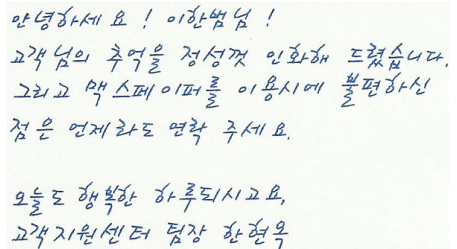
(1) 사람들이 가지는 일반적인 쓰기 특성을 독립적으로 추출하여 모사가 가능하였다. 특히 최초에 글을 작성했던 본인의



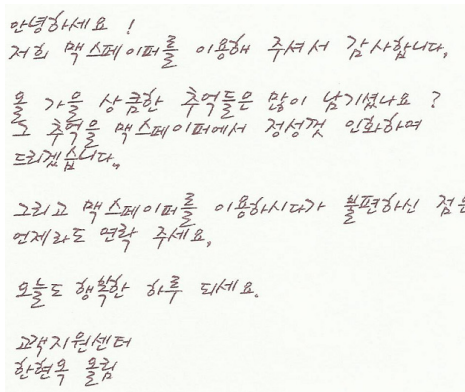
(a) Actual human writing



(b) Simulation output 1



(c) Simulation output 2



(d) Simulation output 3

Fig. 14 Comparison of writings by human or computer

글씨를 흉내 내어 작성할 수 있을 뿐만 아니라 인위적으로 글씨체를 변형하여 작성할 수 있음을 확인하였다.

(2) 사람글씨를 모사하는 것이 가능하지만, 사람의 실제의 글씨는 모눈종이 위에 쓰는 것이 아니기 때문에 글씨가 조금씩 삐뚤삐뚤 해지어, 사람의 필기에 가깝게 보이도록 하는 현상도 가능함을 확인하였다. 그러나 모사 시스템은 컴퓨터가 구동 모터를 조정하면서 볼펜을 쥐 로봇 손이 출력하는 것이기 때문에, 글씨가 줄이 바뀌더라도 모두 평행한 모습을 보여주고 있다. 따라서 글씨 쓰는 로봇 손의 좌우 텀링을 주는 것뿐만 아니라 상하로 위치 변경을 주면서 로봇

손이 종이위에 쓰는 것의 최적변수 설정에 대한 연구가 추가적으로 필요하다.

후 기

이 연구는 중소기업청지원 산학연 공동기술개발사업에 의하여 지원되었음.

References

- (1) Whithaus, C., Harrison, S. B., and Midyette, J., 2008, "Keyboarding Compared with Handwriting on a High-stakes Writing Assessment: Student Choice of Composing Medium, Raters' Perceptions, and Text Quality," *Assessing Writing*, Vol. 13, pp. 4~25.
- (2) Ruirui, Z., and Cheng, C., 2012, "Ink Animation Art in the Digital Age," *Energy Procedia*, Vol. 17, pp. 1295~1298.
- (3) Basteris, A., Bracco, L., and Sanguineti, V., 2012, "Robot-assisted intermanual transfer of handwriting skills," *Human Movement Science*, Vol. 31, pp. 1175~1190.
- (4) Potkonjak, V., Kostic, D., Tzafestas, S., opovic, M., Lazarevic, M., and Djordjevic, G., 2001, "Human-like Behavior of Robot Arms: General Considerations and the Handwriting Task - Part II: the Robot Arm in Handwriting," *Robotics and Computer Integrated Manufacturing*, Vol. 17, pp. 317~327.
- (5) Pérez-Marcosa, D., Buitrago, J. A., and Velásquez, F. D. G., 2011, "Writing through a robot: A proof of concept for a brain-machine interface," *Medical Engineering & Physics*, Vol. 33, pp. 1314~1317.
- (6) Kim, M., and Kim, S., 2012, "Development of a Fan Simulator Using Supercomputer," *J. of the KSMTE*, Vol. 21, No. 5, pp. 805~813.
- (7) Kim, H., 2010, "Development of a New 6-DOF Parallel - type Motion Simulator," *J. of the KSMTE*, Vol. 19, No. 2, pp. 171~177.
- (8) Chang, Y., Liao, C., and Chieng, W., 2009, "Optimal Motion Cueing for 5-DOF Motion Simulations via a 3-DOF Motion Simulator," *J. Control Engineering Practice*, Vol. 17, pp. 170~184.
- (9) Hong, M., and Kim, J., 2009, "A Study on the Motion Base Control by Using Maya," *J. of the KSMTE*, Vol. 18, No. 4, pp. 423~429.
- (10) Rodriguez-Serrano, and Perronnin, J. A. F., 2012, "Synthesizing queries for handwritten word image retrieval," *Pattern Recognition*, Vol. 45, pp. 3270~3276.
- (11) Jawahar, C. V., Balasubramanian, A., Meshesha, M., Namboodiri, A. M., 2009, "Retrieval of online handwriting by synthesis and matching," *Pattern Recognition*, Vol. 42, pp. 1445~1457.
- (12) Hwang, U., and Cho, S., 2004, "Motion Control of a Pneumatic Servo XY-Plotter using Neural Network," *J. of the KSME*, Vol. 28, No. 5, pp. 603~609.