
판서 위치와 메뉴 사용 빈도를 고려한 전자 칠판용 판서 소프트웨어의 동적 메뉴 배치

정시식*, 황민태*

*창원대학교 대학원 정보통신공학과

A Dynamic Menu Layout of the Board Writing Software for IWB system
considering the Writing Position and the Frequency of Menu Usage

Si-Sik Jeong* · Min-Tae Hwang*

*Dept. of Information and Communication Engineering, Changwon National University

E-mail : jsslsy@hanmail.net

요 약

2000년대 초반에 e-러닝 산업이 크게 발전하면서 초·중·고 및 대학교에 이르기까지 전자칠판시스템이 널리 보급되어 스마트 교육 환경이 구축 활용되고 있다. 전자칠판(IWB, Interactive White Board) 시스템은 자체적으로 판서 소프트웨어를 포함하고 있으며, 강사는 이러한 판서 기능을 이용해 전자칠판 상에 필기를 함으로써 교육 효과를 더 높일 수가 있다.

본 논문에서는 사용자의 활용 빈도가 높은 메뉴 버튼들만 화면상에 표출되고, 사용자의 판서 위치에 따라 이들 메뉴 배치가 동적으로 이동하는 판서 소프트웨어의 동적 메뉴 배치 기술에 대해 연구하였다. 이러한 기술을 적용한 판서 소프트웨어 구현 결과 불필요한 사용자의 이동이나 화면 가림 없이 편리하게 판서 메뉴를 이용할 수 있어 향후 스마트 교육의 발전에 크게 기여할 것으로 기대된다.

ABSTRACT

The smart educational environment using the IWB(Interactive White Board) system has been built and developed since e-learning industry was developed significantly in the early 2000's. Basically the IWB system includes the board writing software, and instructors can further increase the training effect by handwriting on the IWB.

In this paper we propose new menu layout mechanism of board writing software that only a few menu buttons are displayed based on the frequency of menu usage and the position of menu layout is dynamically moved according to user's writing position. The implementation of our proposed mechanism shows that it is simple and easy to use without user's unnecessary movement. Therefore it is expected to contribute greatly to the development of a future smart education.

키워드

전자칠판(IWB) 시스템, 판서 소프트웨어, 스마트 교육, 메뉴 사용 빈도, 동적 메뉴 배치

1. 서 론

2000년대 초반에 e-러닝 산업이 크게 발전하면서 초·중·고 및 대학교에 이르기까지 전자칠판(IWB, Interactive White Board) 시스템이 널리 보급되어 스마트 교육 환경이 구축 활용되고 있다[1-2]. 전자칠판 시스템은 자체적으로 판서 소프트웨어를 포함하고 있으며, 강사는 이러한 판서 기능을 이용해 전자칠판 상에 필기를 함으로써 교육 효과를 더 높일 수가 있다. 하지만 대부분의 판서 소프트웨어 메뉴 배치가 전자칠판 화면의 하단에 고정되어 있어 강의 도중에 이를 선택하기 위한 불필요한 움직임을 필요로 하며, 이로 인한 화면 가림으로 학생들의 집중력을 떨어뜨리는

불편함을 초래하고 있다. 아울러 대부분의 판서 소프트웨어는 (그림 1)에서 보는 바와 같이 기본 판서 기능 이외에 부가 기능을 가진 많은 기능 버튼들이 동시에 배치되어 있어 화면 공간의 활용 제약과 더불어 사용자의 혼란을 야기하는 문제점을 가지고 있다[3-4].

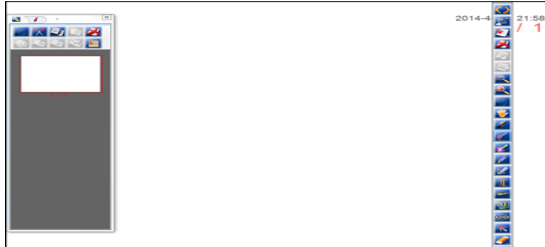


그림 1. 많은 기능 버튼을 가진 판서소프트웨어

이러한 문제를 해결하기 위해 본 논문에서는 사용자의 활용 빈도가 높은 메뉴 버튼들만 화면상에 표출되고, 사용자의 판서 위치에 따라 이들 메뉴 배치가 동적으로 이동하는 판서 소프트웨어의 동적 메뉴 배치 기술에 대해 연구하였다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 제안하는 동적 메뉴 배치 알고리즘을 소개하며, 3장에서는 이러한 알고리즘을 반영한 판서 소프트웨어의 구현 결과를 기술하며, 마지막으로 4장에 결론을 기술하였다.

II. 동적 메뉴 배치 알고리즘

2.1 사용 빈도를 고려한 버튼 배치

본 논문에서는 메뉴의 사용 빈도를 측정하여 빈도가 높은 메뉴 버튼만을 전자칠판 화면에 표시하고, 나머지 메뉴 버튼에 대해서는 확장/축소 버튼을 통해 선택 지원하도록 하였다.

(그림 2)는 본 논문에서 제안하는 사용 빈도에 따른 메뉴 버튼 재배치 알고리즘에 대한 흐름도이다.

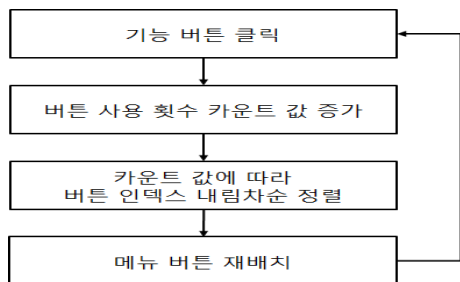


그림 2. 사용 빈도에 따른 버튼 재배치 흐름도

사용자가 메뉴 버튼을 터치함에 따라 카운트 값이 증가하게 되고, 이에 따라 전체 메뉴 버튼의 카운트 값을 내림차순으로 정렬해 상위 메뉴 버튼만 화면상에 재배치하게 된다. [표 1]은 이러한

기능을 구현한 의사코드를 보여준다.

표 1. 버튼 재배치 동작 의사코드

```

void function RearrangeButton
MenuButton[TheNumerOfButton]; // Button Array
TempArray; // Temporary
For (i = 1; i <= NumberOfButton; i++) {
    TempArray[i] = UseCountingArray[i];
    ButtonIndex[i] = i;
}
For (i = 1; i <= NumberOfButton; i++) {
    For (j = i + 1; j <= NumberOfButton; j++) {
        If ( TempArray[j] > TempArray[i] ) {
            Temp1 = TempArray[j];
            TempArray[j] = TempArray[i];
            TempArray[i] = Temp1;

            Temp2 = ButtonIndex[j];
            ButtonIndex[j] = ButtonIndex[i];
            ButtonIndex[i] = Temp2;
        }
    }
}
For (i = 1; i <= NumberOfButton; i++) {
    MenuButtonArrange(MenuButton[ButtonIndex[i]]);
}
    
```

MenuButton은 판서 소프트웨어가 가지고 있는 기능 버튼들을 배열 형태로 정의한 것이고, NumberOfButton은 버튼의 개수를 의미한다. UseCountingArray는 각 버튼의 사용 횟수를 저장하는 배열을 의미하고, ButtonIndex는 사용빈도가 높은 순으로 MenuButton 배열의 index를 저장하는 배열을 의미한다. UseCountingArray는 각 버튼의 사용 횟수를 MenuButton 배열과 동일한 index에 저장하기 때문에 순서가 바뀌면 안 된다. 따라서 내림차순 정렬을 위해 임시 배열인 TempArray와 임시 값인 Temp1, Temp2를 사용하였다. 어떤 버튼을 사용하면 RearrangeButton() 함수가 실행되고, 사용 횟수를 임시 배열에 저장한다. 임시 배열이 버튼의 사용 횟수에 따른 내림차순 정렬을 통해 순서가 바뀌는 경우 ButtonIndex 배열도 사용 횟수가 많은 순서로 버튼의 index 값이 정렬된다. ButtonIndex 배열의 정렬이 끝나면 MenuButtonArrange() 함수를 호출하고 사용 빈도가 높은 순으로 메뉴에 버튼을 배치한다.

2.2 판서 위치를 기반으로 한 메뉴 배치

본 논문에서는 사용 빈도를 고려한 메뉴 버튼 배치 알고리즘과 더불어 사용자의 판서 위치에 따라 메뉴 배치가 자동적으로 이동하는 동적 메뉴 배치 알고리즘을 추가로 제안하였다[5].

대부분의 전자칠판시스템용 판서 소프트웨어에서는 사용자가 전자칠판 화면을 터치하였을 때 터치이벤트가 발생하고 판서 소프트웨어에서는 해당하는 좌표를 획득하여 판서를 표현하게 된다.

이 터치이벤트에서 획득한 좌표를 이용하여 사용자의 현재 판서 위치를 분석하고, 이 때 사용자의 판서 위치가 많이 이동하였다면 메뉴를 사용자에게 가까운 곳으로 자동으로 이동시켜 준다.

(그림 3)은 본 논문에서 제안하는 사용자의 판서 위치에 따라 동적으로 메뉴가 배치되는 알고리즘의 흐름도를 보여준다.

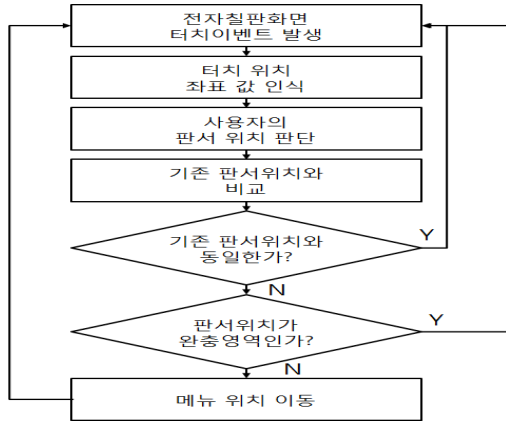


그림 3. 동적 메뉴 배치 기능의 흐름도

사용자의 판서 위치는 (그림 4)에서 보는 바와 같이 전자칠판 화면을 총 4개의 영역으로 분할하여 구분하였으며, 판서 위치에 따라 메뉴 버튼들이 해당 영역의 좌우 가장자리에 동적으로 배치 되도록 하였다.

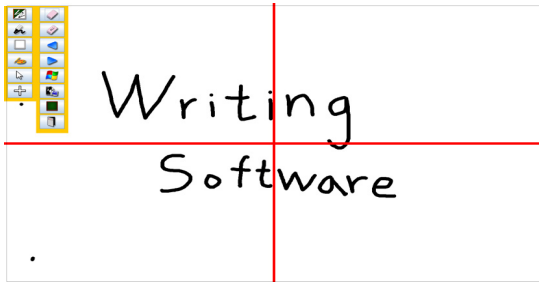


그림 4. 4개의 영역으로 분할한 전자칠판 화면

만약에 사용자가 영역 경계면을 넘나들면서 판서를 할 경우에는 잦은 메뉴 이동이 발생해 혼란을 초래할 수 있으므로 이를 해결하기 위해 (그림 5)와 같이 완충 영역을 설정해 판서 도중에 완충 영역을 벗어났을 때에만 메뉴 버튼들이 이동할 수 있도록 구현하였다.



그림 5. 분할된 영역의 완충 영역

[표 2]는 사용자의 판서 위치에 따라 메뉴가 이동하는 기능을 구현한 의사코드를 보여준다.

표 2. 동적 메뉴 배치 동작 의사코드

```
void function TouchEvent()
x = X_TouchPosition; // X-coordinate on Screen
y = Y_TouchPosition; // Y-coordinate on Screen

If (!(x > (ScreenWidth / 2 - X_BufferArea))
    && (x < (ScreenWidth / 2 + X_BufferArea)))
    || ((y > (ScreenHeight / 2 - Y_BufferArea))
        && (y < (ScreenHeight / 2 + Y_BufferArea)))
    {
        If ((x <= (ScreenWidth / 2 )) &&
            (y <= (ScreenHeight / 2))) {
            MenuLayout(0, 0);
        }
        Else If ((x >= (ScreenWidth / 2 )) &&
            (y <= (ScreenHeight / 2))) {
            MenuLayout(ScreenWidth - MenuWidth, 0);
        }
        Else if ((x <= (ScreenWidth / 2 )) &&
            ((y >= (ScreenHeight / 2))) {
            MenuLayout(0, ScreenHeight - MenuHeight);
        }
        Else if ((x >= (ScreenWidth / 2 )) &&
            ((y >= (ScreenHeight / 2))) {
            MenuLayout(ScreenWidth - MenuWidth,
                ScreenHeight - MenuHeight);
        }
    }
}
```

X_TouchPosition과 Y_TouchPosition은 각각 현재 사용자가 판서를 위해 터치하고 있는 전자칠판 화면의 x 및 y 좌표를 의미한다. ScreenWidth와 ScreenHeight는 각각 전자칠판 화면의 넓이와 높이를 의미하며, 전자칠판 화면을 좌표평면과 같이 4개의 영역으로 분할하기 위해 사용된다.

X_BufferArea와 Y_BufferArea는 사용자가 4개로 분할된 영역의 경계면에서 판서를 할 경우 메뉴의 잦은 이동이 발생하므로 이를 방지하기 위한 완충영역을 의미한다.

사용자가 판서를 위해 전자칠판 화면을 터치하게 되면 TouchEvent()함수가 실행되고, 현재 판서 위치의 x 및 y 좌표를 가져오게 된다. 그리고 현재 판서 위치가 완충 영역이 아닐 경우에 전자칠판 화면의 크기를 이용하여 판서 위치가 4개로 분할된 영역 중에 어느 영역에 속하는지 판단하고, MenuLayout(x,y) 함수를 이용하여 전자칠판 화면에 현재 판서 중인 영역의 좌우 가장자리로 메뉴를 배치하게 된다.

III. 동적 메뉴 배치 기능을 가진 판서 소프트웨어의 구현

3.1 소프트웨어 개발 환경

전자칠판시스템용 판서 소프트웨어는 자바 프로그래밍(Java Programming)을 이용해 개발하였으며, 개발 툴로는 자바 소프트웨어 개발에 적합한 Eclipse Indigo 버전을 사용하였다. 그리고 50인치 대형 터치모니터를 PC에 연결해 전자칠판 역할을 대신하도록 구현하였다.

표 3. 소프트웨어 개발 환경

개발 언어	Java
개발 툴 및 버전	Eclipse Indigo
PC 사양	Intel Core i5-2320 CPU 3GHz
전자칠판 사양	MIT-LED 50 inch
.ppt 포맷 지원 라이브러리	Apache POI API
.pdf 포맷 지원 라이브러리	Apache PDFBox Library

3.2 구현 결과

(그림 6)은 본 논문에서 직접 구현한 전자칠판 시스템용 판서 소프트웨어를 보여준다.



그림 6. 전자칠판시스템용 판서 소프트웨어

여기서는 메뉴 버튼의 사용 빈도에 따른 메뉴 배치 알고리즘만 적용한 경우이다. 그림에서 살펴보는 바와 같이 강의 진행에 필수적이면서 활용 빈도가 높은 메뉴 버튼들만 자그마한 크기로 가장자리에 배치되어 있어 대부분의 화면을 판서 작업에 활용할 수 있는 장점과 더불어 단순하면서도 효율적인 특징이 있다.

(그림 7)은 본 논문에서 구현한 확장 버튼을 이용한 메뉴 확장 기능을 보여주고 있다.

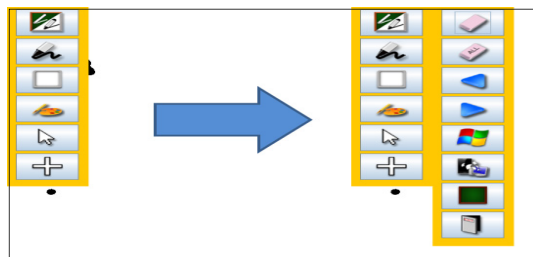


그림 7. 메뉴 확장 기능

그림에서 보는 바와 같이 구현한 판서 소프트웨어의 메뉴 버튼은 기본적으로 사용 빈도가 높은 5개의 메뉴 버튼만을 보여주고 있다.

강의 도중에 다른 기능의 메뉴 버튼을 사용하기 위해서는 '+' 모양의 확장버튼을 누르면 된다. 그러면 기본 메뉴 버튼들 옆에 확장 메뉴 버튼들이 나타나게 된다.

본 논문에서는 메뉴 버튼의 사용 빈도에 따른 메뉴 배치 알고리즘에 추가하여 사용자의 판서 위치에 따라 메뉴 버튼들이 동적으로 이동하여

사용자의 편의성을 높이는 기능을 추가로 구현하였다. (그림 8)은 전자칠판 화면을 4개의 영역으로 분할하여 사용자의 판서 위치에 따라 메뉴 버튼들이 각 화면의 좌우 가장자리 부분으로 자동으로 이동하는 모습을 보여주고 있다.



그림 8. 판서 위치에 따른 메뉴의 이동

IV. 결 론

본 논문에서는 사용자의 판서 위치와 메뉴 버튼의 사용 빈도를 고려한 판서 소프트웨어의 동적 메뉴 배치 알고리즘에 대해 연구하였다.

제안 기술은 사용자의 메뉴 사용 빈도를 측정해 사용 빈도가 높은 메뉴 버튼들만 화면상에 배치하고 나머지 메뉴는 확장 버튼을 두어 지원한다. 그리고 전자칠판 화면을 4개의 영역으로 나누고서 영역 경계면에는 완충 구간을 두어 사용자의 판서 위치에 따라 메뉴 배치가 자동적으로 이동하게 된다. 이러한 기술을 적용한 판서 소프트웨어 구현 결과 불필요한 사용자의 이동이나 화면 가림 없이 편리하게 판서 메뉴를 이용할 수 있어 향후 스마트 교육의 발전에 크게 기여할 것으로 기대된다.

참고문헌

- [1] 박용서, 고상숙, “수학수업에서 저비용으로 구성된 전자칠판의 활용효과에 대한 연구”, 한국학교수학회논문집, 제14권, 제1호, pp. 1-29, 2011.
- [2] 이성숙, “실과 가정영역에서 스마트교육 활용 방안”, 한국실과교육연구학회 논문집, 제19권, 제1호, pp. 53-72, 2013.
- [3] 설문규, 손창익, “초등학교에서 스마트 교육에 대한 교사들의 활용인식 조사”, 한국정보교육학회논문지, 제16권, 제3호, pp. 309-318, 2012.
- [4] 이태곤, 이지연, “예비교사의 전자칠판 사용 및 추천의도에 영향을 미치는 요인”, 인하대학교 교육연구소 논문집, 제19권, 제2호, pp. 63-97, 2013.
- [5] 정진욱, 최윤수, 진교홍, 황민태, “사용자 동작 인식을 지원하는 판서 소프트웨어 개발”, 한국정보통신학회 논문지, 제19권, 제5호, pp. 1031~1037, May 2015.