

이미지처리 기법을 이용한 Grid Logic 게임 콘텐츠 개발

오갑석*

요약

최근 인터넷의 발달과 더불어 다양한 아케이드 게임이 네트워크를 통하여 제공되고 있으며, 누구나 쉽게 접할 수 있는 Grid Logic 게임의 경우 제공되는 퍼즐 이외에는 게임을 할 수 없다는 단점이 있다. 이를 개선하기 위하여 본 논문에서는 이용자의 이미지를 사용할 수 있는 Grid Logic 게임 콘텐츠를 개발하였다. 이를 위하여 영상처리의 전처리 단계인 임계치 결정방법을 제안하고, 제안 방법으로 구현된 이진 이미지에서 목표물을 검출하는 방법과 게임 데이터로 변환하는 방법을 제시하였으며, 게임을 완료한 시점에는 특정 형상으로 의미 전달이 가능하도록 하였다. 문자, 로고, 인물 등의 목표물을 포함하는 10종의 이미지에 제안 알고리즘을 적용하여 JAVA 애플릿으로 구현하고, 실험을 통하여 제안 방법이 Grid Logic 게임의 데이터 획득에 적합함을 보인다.

A Development of Grid Logic Game Contents by Using Image Processing Method

kabsuk Oh

Abstract

Recently, various kinds of arcade games are offered through the network with the internet's development. And for the Grid Logic game, it is opened up for everyone who uses the internet but it has a disadvantage that only the provided puzzles can be played. To improve this, in this paper, we developed a Grid Logic game contents using an image of user's as a puzzle. In order to do this, we suggested a threshold decision method, the pre-processing stage of image processing. We showed a method of detecting aim image from a binary image, showed up by the suggested way, and a method of changing into the game data and carrier of the meaning as a specific image at the end of the game are the objects of this paper. The suggested algorithm is constructed as a Java applet and applied to the 10 objects such as characters, logos, persons, etc. to show that this algorithm is suitable for the appropriate acquisition of the Grid Logic game data through the experiment.

Keywords : Image processing, Binarization, Threshold, Game Contents

1. 서론

근래의 컴퓨팅 환경은 인터넷 기술의 보급과 컴퓨터의 성능 향상으로 인해 컴퓨터의 응용분야 확산을 가속화 시키고 있다. 특히 고속 인터넷의 확산으로 전문기술 영역뿐만 아니라 오락과 여가 영역에서도 컴퓨터의 활용이 매우 쉽게 이루어진

다. 한국의 인터넷 인프라 발전으로 인터넷 게임 사용자는 날이 증가하고 있으며, 수토쿠 퍼즐 [1], 미로게임[2] 등을 비롯하여 양질의 게임 콘텐츠 개발 관련 연구[3]가 이루어지고 있다.

인터넷 게임은 다양한 장르로 분류되는데, 본 연구의 대상인 Grid Logic 게임은 아케이드 장르로서 인터넷을 통하여 제공되는 프로그램을 클라이언트 환경에서 실행하는 것으로 데이터는 DB, Text 파일 또는 배열 형태로 제공된다. 게임은 제공된 데이터인 힌트수에 의해 특정 형상의 퍼즐을 완성하는 것이지만, 때로는 특정 형상이 될 것이라 유추하는 것도 문제해결에 도움이 된다. 그

※ 제일저자(First Author) : 오갑석
접수일:2009년 08월 22일, 완료일:2009년 9월 22일
* 동명대학교 정보통신대학 정보통신공학과
oks@tu.ac.kr

러나 사용자만이 사용할 수 있는 임의의 게임 데이터를 입력하여 게임을 할 수 있는 방법은 제공되지 않고 있다. 이러한 문제를 해결하기 위한 하나의 방안으로 사용자가 페인팅 스타일 등의 에디터로 데이터를 입력하는 방법을 고려할 수 있다. 이 경우에는 사용자가 어떤 특정 의미를 갖는 픽셀 단위의 데이터 구성을 알고 있으므로 게임으로서의 흥미가 떨어지며, 사용자는 이러한 데이터 가공에 상당한 시간을 소비하게 된다. 이에 본 연구에서는 사용자의 데이터 가공 능력과 무관하게 사용자가 가지고 있는 이미지 파일만을 선택하면 결과를 예측하기 쉽지 않은 데이터를 생성하여 게임에 적용하는 방법을 제시하고자 한다.

한편 이미지는 많은 정보를 내포하고 있어, 가공 방법에 따라 다양한 형태의 의미 있는 데이터로 변환할 수 있다. 영상 인식 과정에서 사용되는 이미지의 특징이 다르기 때문에 이진화(binanzation)를 위한 임계치(threshold) 선택은 중요한 문제로서 다양한 방법들이 제시되어 있다. 실험적 경험치를 임계치로 결정하는 방법[4][5]은 피부색 검출과 같은 특징을 찾는데 적합하고, 목적함수의 오차 제곱합 가중치를 최소화하는 방법[6], 구간 분할 및 통합 방법[7] 및 뉴럴 네트워크와 퍼지제어를 융합한 방법[8]은 반복 연산으로 처리 시간이 많이 소요된다.

이러한 문제점을 해결하기 위하여 본 연구에서는 임계치 선택의 대표적인 방법인 p-tile 임계치, 히스토그램 골을 임계치로 설정하는 방법 및 분할된 영역의 분포에 따라 임계치를 결정하는 방법에 대하여 고찰하고, 빠른 계산 시간을 고려함과 동시에 목표물의 범위를 좁히는 새로운 임계치 결정방법을 제안한다. 이어서 목표물 추출과 게임 데이터 변환 방법을 제시한다. 또한 JAVA 애플릿으로 게임 시스템을 구축하고, 실험을 통하여 제안 알고리즘의 유효성을 검증한다.

본 논문의 구성은 다음과 같다. 2장에서는 Grid Logic 게임에 적용하는 임계치 결정 알고리즘을 제안하고, 3장에서는 이진화 방법에 제안 알고리즘을 적용하여 목표 이미지를 추출하는 방법과 획득된 데이터를 게임에 적용하는 방법을 제시한다. 4장에서는 실험 방법 및 검토를 통하여 제안 방법의 유효성을 검증하고, 5장에서 결론을 맺는다.

2. 임계치 결정 방법의 기존연구 및 제안 방법

본 장에서는 Grid Logic의 원리를 설명하고, 이미지의 이진화에 필요한 기존의 임계치 결정 방법에 대한 장단점을 소개하고, Grid Logic 게임에 적용하는 새로운 알고리즘을 제시한다.

Grid Logic은 1988년 5월에 일본인 테즈야 니시오에 의해 처음 창안된 퍼즐로서 노노그램, 로직퍼즐 등으로도 불리고 있다[9]. (그림 1)은 게임 도중의 모습을 나타내었다. 게임 시작과 동시에 빈 바둑판 모양의 좌측과 상측에 가로 및 세로 힌트 숫자가 제공된다. 게임 방법은 힌트 숫자를 논리적으로 판단하여 네모 칸에 숨겨진 데이터의 유무를 선택하는 것이다. 예를 들어 (그림 1)의 5행에 있는 2, 4, 2의 힌트수는 가로로 2개의 연속점과 4개의 연속점 및 2개의 연속점이 있으며, 각 연속점 사이에는 1개 이상의 불연속점이 존재한다는 의미이다. 특정 칸에서 마우스의 오른쪽 버튼과 왼쪽 버튼을 선택하면 데이터의 유무를 표시할 수 있다. 따라서 Grid Logic 게임을 구현하려면 0과 1의 값을 갖는 일련의 데이터가 요구된다.

			2	4							
	1	2	1	1	10	9	4	2	2	1	
2											
4											
6											
8											
2 4 2			X					X			
2											
2											
2											
1 2											
2											

(그림 1) Grid Logic의 예

한편, 임계치 이진화 방식은 식(1)과 같이 대상 이미지 $f(x, y)$ 의 화소값이 T보다 크면 255로, 그렇지 않은 화소들은 0으로 분리한다.

$$g(x,y) = \begin{cases} 255 & \text{if } f(x,y) > T \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

이와 같은 임계치 이진화는 고정적인 임계치를 이용하면 쉽게 처리할 수 있다. 그러나 영상 인식 과정에서 사용되는 이미지의 특징이 다르기 때문에 임계치를 다양한 방법으로 선택하여 결정하는 방법은 다음과 같다.

2.1 p-tile 임계치

p-tile 임계치는 밝은(또는 어두운) 영역의 비율을 이용하여 임계치를 결정하는 방법[10][11]으로, 이미지의 밝기 분포를 계산하고 설정된 비율에 맞는 임계치를 찾아서 이진화 한다. 이 방법은 계산이 간단하지만, 이미지 내에서 물체의 영역이 차지하는 비율을 입력해야하는 단점이 있다.

2.2 히스토그램의 골을 임계치로 설정

히스토그램의 골을 찾아 임계치로 설정하는 방법은 히스토그램의 두 봉우리(peak) 사이의 골을 임계값으로 설정하는 방법이다[10,11]. 히스토그램의 골을 찾는 방법은 첫 번째의 가장 높은 봉우리를 먼저 찾고, 다른 봉우리를 찾아 두 봉우리 사이의 골을 찾는다. 구체적인 알고리즘은 다음과 같다.

Step 1. 이미지 $f(x, y)$ 의 k 밝기에 대한 빈도수 (Hist[k])를 구한다.

Step 2. 히스토그램(Hist[k])의 최대값 위치(M_1 , 첫 번째 봉우리)를 찾는다.

Step 3. 최대값 위치(M_1)를 제외한 모든 위치에서 수식(2)를 만족하는 두 번째 봉우리(M_2)를 찾는다.

$$M_2 = \max_k(|k - M_1| \text{Hist}[k]) \quad (2)$$

Step 4. 식(3)을 이용하여 두 봉우리 사이에서 골을 찾아 최종 임계치로 결정한다.

$$T = (M_1 + M_2)/2 \quad (3)$$

이 방법은 계산이 간단하지만, 히스토그램이 양봉의 특징을 갖지 않을 경우 적용이 곤란하다.

2.3 분할된 영역의 분포에 따른 임계치 결정

분할된 영역의 분포에 따라서 임계치를 결정하는 방법은 임계치를 기준으로 분할된 영역이 최적의 임계치인가를 반복적으로 변경하여 임계치를 찾는 방법이다[10,11]. 히스토그램에서 초기 임계치를 설정하고, 임계치를 기준으로 분할된 두 개의 영역(cluster)의 중심(m_1, m_2)을 찾고, 두 영역의 중심을 새로운 임계치(T_i)로 결정한다. 이와 같은 임계치는 반복적으로 계산하고, 이전의 임계치와 새로운 임계치가 동일하면 최적의 임계치로 지정하여 반복을 멈춘다. 구체적인 알고리즘은 다음과 같다.

Step 1. 이미지 $f(x, y)$ 의 k 밝기에 대한 빈도수 (Hist[k])를 구한다.

Step 2. 초기 임계치(T_0)를 수식(4)과 같이 결정한다.

$$T_0 = (M_1 + M_2)/2 \quad (4)$$

단, M_1 은 빈도수가 0보다 큰 최소 화소이고, M_2 는 빈도수가 0보다 큰 최대 화소이다.

Step 3. 임계치를 경계로 식(5)와 같이 두 영역의 중심(m_1, m_2)을 찾는다.

$$\begin{pmatrix} m_1 = \frac{1}{N_1} \sum_{j=M_1}^{T_i-1} j \times \text{Hist}[k], \\ m_2 = \frac{1}{N_2} \sum_{j=T_i}^{M_2-1} j \times \text{Hist}[k] \end{pmatrix} \quad (5)$$

여기서 N_1 과 N_2 는 각 영역에 포함된 화소의 합이다.

Step 4. 두 중심 간의 중간 위치를 수식(6)을 이용하여 새로운 임계치(T_i)로 지정한다.

$$T_i = (m_1 + m_2) / 2 \quad (6)$$

Step 5. $T_i \neq T_{i-1}$ 이면 Step 3으로 돌아가서 반복하고, 같으면 식(6)의 T_i 를 최적의 임계치로 지정하고 종료한다.

이 방법은 히스토그램의 골을 찾는 방법에 비해 적절한 임계치를 결정할 수 있으나 반복적 계산으로 인하여 처리 시간이 많이 소요된다.

2.4 제안하는 임계치 결정 방법

Grid Logic 게임의 데이터 획득에서 빠른 계산 시간을 고려하면, 뚜렷한 목표물 이외의 이미지는 잡음으로 간주하여 임계치 결정 단계에서 제거할 필요가 있다. 따라서 기존 방법의 장점인 계산량이 적고, 목표물의 범위를 좁힐 수 있는 임계치 결정 방법을 제시한다. 제안 알고리즘은 지정 구간의 최대 히스토그램 위치와 초기 임계치 사이의 최소값을 최적의 임계치로 결정하는 것으로 구체적인 내용은 다음과 같다.

Step 1. 이미지 $f(x, y)$ 의 k 밝기에 대한 빈도수 (Hist[k])를 구한다.

Step 2. 초기 임계치(T_0)를 수식(7)과 같이 설정한다.

$$T_0 = (M_1 + M_2) / 2 \quad (7)$$

단, M_1 은 빈도수가 0보다 큰 최소 화소이고, M_2 는 빈도수가 0보다 큰 최대 화소이다.

Step 3. 구간 $[M_1, T_0]$ 의 최대 히스토그램 (Hist[k])값의 위치(H_0)를 찾는다.

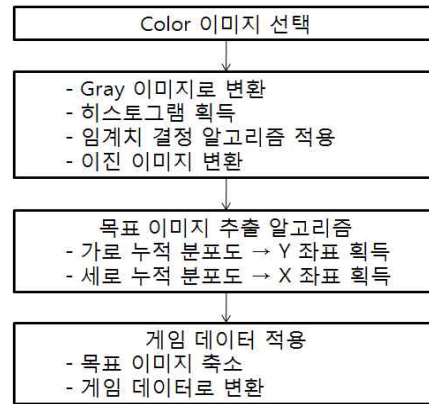
Step 4. 수식(8)과 같이 H_0 와 초기 임계치(T_0) 사이의 최소값을 최적의 임계치로 지정하고 종료한다.

$$T = \min_{k=H_0}^{T_0} (Hist[k]) \quad (8)$$

3. 게임데이터 획득 알고리즘

게임 사용자가 가지고 있는 이미지를 게임 데이터로 사용하기 위하여 이미지를 이진화하여 Grid Logic 게임의 데이터 형식인 0과 1로 구성된 2차원 배열 형태로 제공해야 한다.

본 연구에서는 칼라 이미지를 흑백 이미지(gray level)로 변환하여 밝기 값의 히스토그램을 구하고 제안한 임계치 결정 알고리즘을 적용하여 이진 이미지(binary image)로 변환한다. 그리고 획득한 이진 이미지의 가로 및 세로 픽셀 분포도로부터 목표 이미지 영역에 해당하는 좌표를 추출하는 방법을 제시한다. 이렇게 추출된 이진 이미지 영역을 게임데이터로 변경하여 적용하는 게임 데이터 획득 알고리즘을 (그림 2)에 나타내었다.



(그림 2) 게임 데이터 획득 알고리즘

3.1 목표 이미지 추출 방법

Grid Logic 게임의 데이터는 사각형의 크기를 갖는다. 따라서 게임 데이터에 적용할 목표 이미지는 변환된 이진 이미지로부터 사각형의 구조를 갖도록 다음과 같이 추출한다.

Step 1. 이진 이미지에 가로 및 세로로 투영하여 검은색 누적픽셀 분포도를 구한다.

Step 2. 각 분포도로부터 처음으로 10이상의 값을 갖는 구간의 좌표{시작좌표(X_0, Y_0), 끝좌표(X_1, Y_1))를 구하고, 좌표 영역 내부를 목표 이미지로 결정한다.

Step 3. 목표 이미지의 크기가 25X25 픽셀 이상의 크기이고, 가로 세로의 비율이 10:6의 범위 이내일 경우, 최종 목표 이미지로 결정하고 종료한다. 그렇지 않을 경우 선택 영역의 이전 이미지를 삭제하고 Step 2를 반복한다.

3.2 데이터 변환 및 적용 방법

이렇게 추출된 이진 이미지 영역의 정보를 게임 데이터로 변환하기 위하여 목표 이미지의 가로 또는 세로의 최대 크기를 25 픽셀이 되도록 영상 축소 기법(subsampling)을 적용하여 일정 크기의 이진 이미지를 확보한다. 추출된 이진 이미지는 0(검정색)과 255(흰색)의 값을 가지므로 게임 데이터용 2차원 배열에 각각 1과 0의 값으로 저장하고, 가로 및 세로로 연속되어 나타나는 검정색의 수를 순차적으로 힌트 수 저장용 배열에 저장하여 Grid Logic 게임의 데이터로 변환한다. 이때 문제 저장용 배열의 크기 인덱스는 목표 이미지의 크기에 의해 결정되는데, 가장 많은 힌트 수가 존재 할 경우를 고려하여 힌트 수 저장용 배열의 크기 인덱스는 문제 저장용 배열의 인덱스가 짝수일 경우 식(9)와 같이 결정하고, 홀수일 경우 식(10)과 같이 결정한다.

$$\text{인덱스} = \text{문제크기}(x \text{ 또는 } y)/2 \quad (9)$$

$$\text{인덱스} = \text{문제크기}(x \text{ 또는 } y) + 1/2 \quad (10)$$

문제 저장용 배열과 힌트 저장용 배열의 크기가 결정되면 문제 배열 값으로부터 힌트의 수를 결정하는 방법은 (그림 3)와 같다.

이상과 같이 구해진 문제 데이터를 게임 콘텐츠로 구현할 때 다음과 같은 사항을 고려한다.

세로 힌트수는 아래쪽 정렬로 문제의 위쪽에 나타내고, 가로 힌트수는 오른쪽 정렬로 문제의 왼쪽에 나타낸다. 문제의 칸은 모두 빈사각형(□)으로 표현하고 왼쪽마우스를 클릭하면 데이터가 '있음'을 표시(■) 하고, 오른쪽 마우스를 클릭하면 데이터가 '없음'을 표시(×)를 한다. 또한 정답 확인을 위하여 정답 저장용 배열과 같은 크기의 정답 확인용 배열을 0으로 초기화하여 미리 준비한다. 사용자가 문제의 칸에 데이터가 '있음'을 선택할 경우에는 해당 배열에 1의 값을 저장하고,

'없음'을 표시할 경우에는 0의 값을 저장 한다. 이때 이미 표시된 칸에 다시 마우스를 클릭하면 지우기, 선택 기능이 반복된다.

```
// 가로 힌트 수 결정 방법
for(i=0; i<y; i++){
    count = 0;    k=0;
    for(j=0; j<x; j++){
        if(Nemo[i][j]==1) count++;
        if(Nemo[i][j]==0 && !(count==0)){
            Nemo_Width[i][k]=count; count=0; k++; }
        }
        if(Nemo[i][x-1]==1) Nemo_Width[i][k]=count;
    }
}
// 세로 힌트 수 결정 방법
for(i=0; i<x; i++){
    count = 0;    k=0;
    for(j=0; j<y; j++){
        if(Nemo[j][i]==1) count++;
        if(Nemo[j][i]==0 && !(count==0)){
            Nemo_Height[i][k]=count; count=0; k++; }
        }
        if(Nemo[y-1][i]==1) Nemo_Height[i][k]=count;
    }
}
```

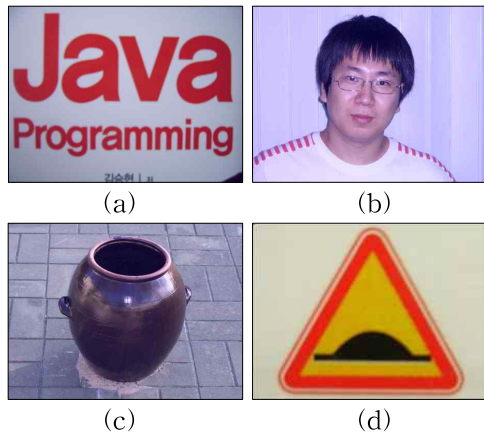
4. 실험 결과

4.1 시스템 구성

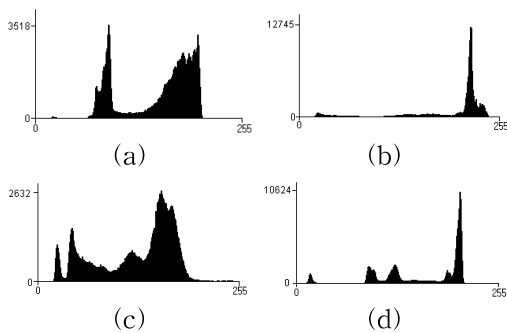
본 논문에서 제안한 임계치 결정 방법과 Grid Logic 게임의 데이터 획득 방법의 유효성을 알아보기 위하여 Intel Pentium Dual CPU 1.60GHz 프로세서, 2GB RAM 시스템 환경에서 JDK 6 버전의 JAVA 애플릿으로 시스템을 구성 하였으며, 디지털카메라로 촬영한 이미지와 웹페이지에서 구한 이미지 및 로고 등을 454×341 픽셀 크기인 JPG 파일로 저장하여 실험에 사용 하였다.

4.2 임계치 적용 실험

(그림 4)에는 책표지의 문자, 사람, 향아리, 도로표지판 등이 게임 데이터로 변환했을 때 의미 있는 대표적인 이미지 4개를 제시하였다. (그림 5)에는 실험에 사용한 4개의 이미지에 대한 히스토그램 분포도를 나타내었다.



(그림 4) 원본 칼라 이미지 : (a) 책표지, (b) 사람, (c) 항아리, (d) 도로 표지판



(그림 5) 히스토그램 : (a) 책표지, (b) 사람, (c) 항아리, (d) 도로 표지판

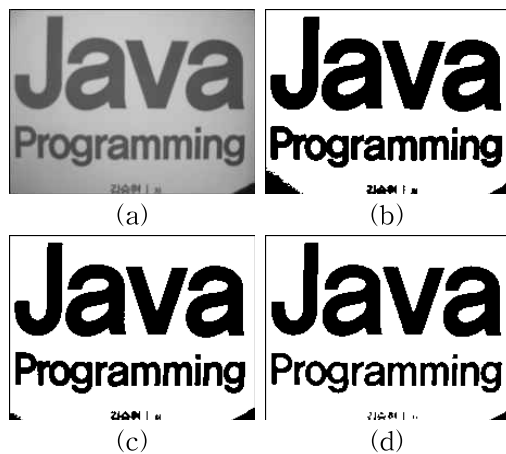
그리고 (그림 4)의 4개의 원본 이미지에 대하여 히스토그램의 골을 찾아 임계치로 설정하는 방법 (기존방법1로 표기), 분할된 영역의 분포에 따른 임계치 결정방법(기존방법2로 표기), 그리고 본 논문에서 제안한 임계치 결정방법(제안방법으로 표기)을 적용하여 구해진 임계치를 <표 1>에 나타내었다.

<표 1> 임계치

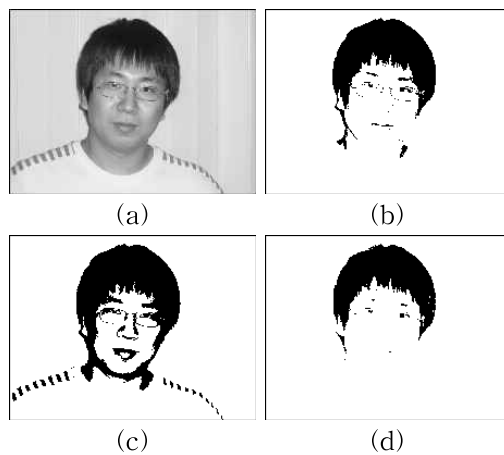
이미지	임계치 선택방법		
	기존방법1	기존방법2	제안방법
책표지	145	133	110
사람	120	145	96
항아리	99	106	93
도로표지판	148	150	109

<표 1>에서 알 수 있듯이 제안방법은 기존의 방법보다 낮은 임계치를 갖는다.

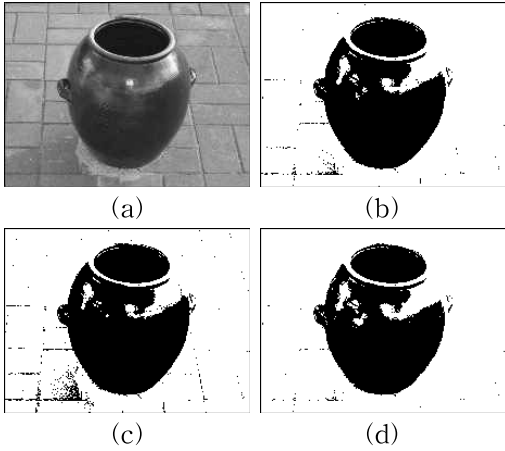
실험에 사용한 4개의 이미지에 대하여 2장에서 제시한 임계치 결정 방법을 적용하고 이진 이미지를 구하여 (그림 6)부터 (그림 9)에 나타내었다. 여기서, 2장에서 제시한 p-tile 임계치 결정법은 물체의 크기를 이미지에 따라 인위적으로 판단하여 제시해야하므로 제외 하였다. 각 그림의 (a)는 흑백 이미지(8비트)로 변환한 것이며, (b)는 기존 방법1, (c)는 기존방법2, (d)는 제안방법을 적용한 결과를 나타내었다. 여기서 각 방법에 적용된 임계치는 <표 1>에서 제시한 값이다.



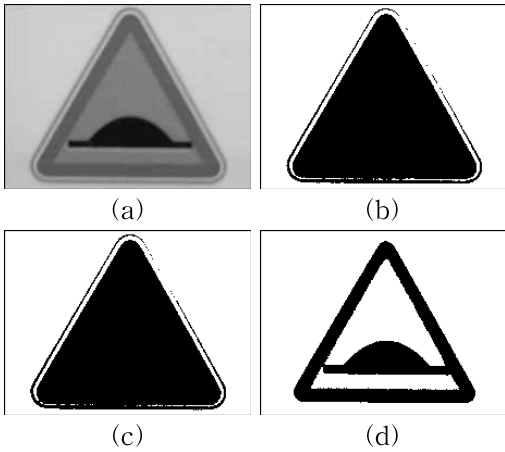
(그림 6) 책표지 이미지에 적용한 결과



(그림 7) 사람 이미지에 적용한 결과



(그림 8) 항아리 이미지에 적용한 결과



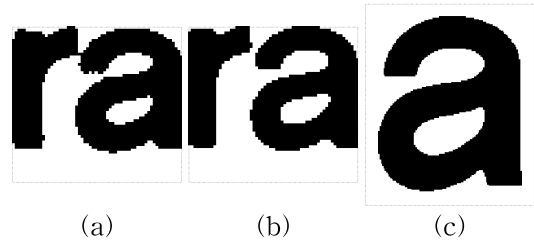
(그림 9) 도로표지판 이미지에 적용한 결과

(그림 6), (그림 7), (그림 8) 및 (그림 9)에 나타난 결과로부터 제안방법이 기존방법보다 뚜렷한 목표물 이외의 이미지는 잡음으로 간주하여 본 단계에서 직관적으로 제거되었음을 알 수 있다.

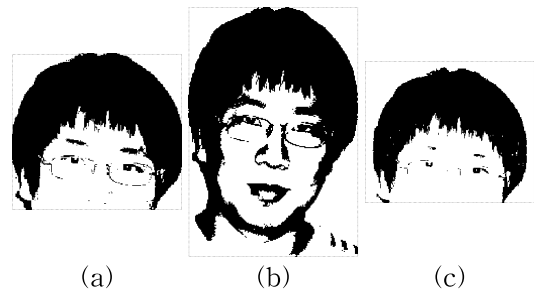
4.3 목표 이미지 추출 및 게임 데이터 적용 실험

목표 이미지 추출 및 게임 데이터 적용 실험은 임계치 결정 방법에 따라 획득한 이진 이미지를 제안한 목표 이미지 추출 방법에 적용하여 실시하였다. 실험에 사용한 4개의 이미지에 대하여 3장에서 제시한 목표 이미지 추출방법에 의한 결과를 (그림 10)-(그림 13)에 나타내었다. 각 그림에서 (a)는 히스토그램의 골을 찾아 임계치로 설

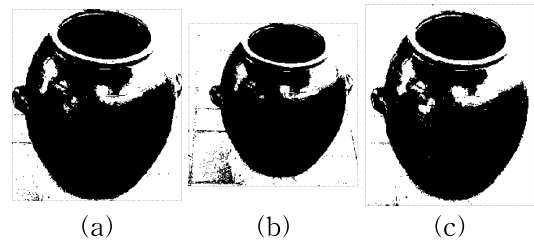
정하는 방법, (b)는 분할된 영역의 분포에 따른 임계치 결정방법, 그리고 (c)는 제안한 임계치 결정방법에 의한 이진 이미지에 적용한 결과이다.



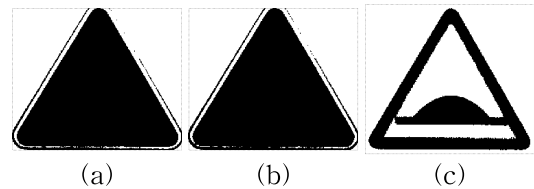
(그림 10) 책표지의 목표 이미지 추출 결과
(a) 기존방법1, (b) 기존방법2, (c) 제안방법



(그림 11) 사람의 목표 이미지 추출 결과
(a) 기존방법1, (b) 기존방법2, (c) 제안방법



(그림 12) 항아리의 목표 이미지 추출 결과
(a) 기존방법1, (b) 기존방법2, (c) 제안방법



(그림 13) 도로표지판의 목표 이미지 추출 결과
(a) 기존방법1, (b) 기존방법2, (c) 제안방법

4개의 이미지로부터 추출한 목표이미지는 모두 원본 이미지의 특정 부분을 원만하게 전달하고 있으며, 그 크기는 제한한 목표이미지 추출 방법의 Step 3의 제한 조건을 만족하고 있다. (그림 10)의 기존방법들에서는 원본 이미지의 'Programming' 단어로부터 획득된 'ra' 문자이며, 제안방법의 결과에서는 'Java' 단어에서 첫 번째 'a' 문자를 추출한 결과이다. (그림 11)과 (그림 12)의 목표 이미지 추출 결과에서는 사람의 얼굴과 향아리의 특징을 표현하고 있다. (그림 13)의 결과에서는 주의표지판임은 직감할 수 있으나 그 내용이 무엇인지는 제안방법에서만 확인할 수 있다. 이러한 결과는 임계치 결정 방법의 결과에서 예상된 것으로, 사용자가 가지고 있는 이미지를 게임데이터로 변환할 수 있음을 보인 것이다.

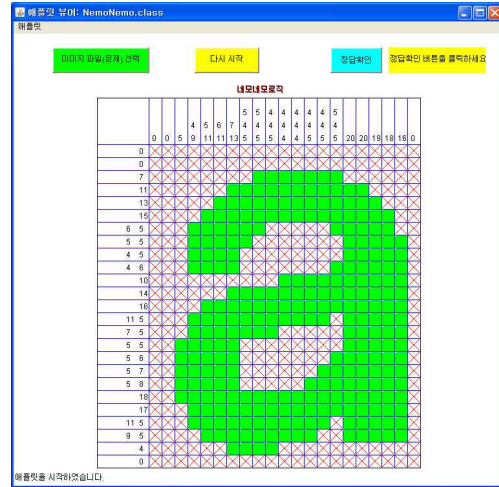
<표 2>는 (그림 5)에 나타낸 목표 이미지의 크기와 게임 데이터의 크기이다.

<표 2> 목표 이미지와 게임 데이터의 크기

이미지	목표 이미지 (게임 데이터)		
	기존방법1	기존방법2	제안방법
책표지	62X59 (25X23)	61X58 (25X23)	128X152 (21X25))
사람	189X178 (25X23)	192X271 (17X25)	208X182 (25X21)
향아리	255X281 (22X25)	306X295 (25X24)	261X298 (21X25)
도로표지판	359X312 (25X21)	395X312 (25X21)	353X317 (25X22)

<표 2>에 보이듯이 추출된 목표 이미지의 가로 또는 세로의 최대 크기는 임계치 결정 방법에 따라 차이가 나지만, 게임에 적용된 데이터의 최대 크기는 설계한 게임 콘텐츠의 제한 조건을 만족하는 25 픽셀로 축소 되었다. 또한 제시하지 않은 6개의 이미지로부터 추출된 목표 이미지의 경우에도 동등한 결과를 얻었다.

본 논문에서 제안한 이미지처리 기법을 적용하여 완성된 Grid Logic 게임 콘텐츠의 예를 (그림 14)에 나타내었다.



(그림 14) 게임 콘텐츠 실행 결과

(그림 14)에 보이듯이 사용자 이미지를 선택할 수 있는 버튼과 다시시작 및 정답확인을 위한 버튼을 준비하였다. 따라서 사용자가 가지고 있는 다양한 형태의 이미지를 선택함으로써 게임을 할 수 있다.

5. 결론

본 논문에서는 사용자가 가지고 있는 이미지를 Grid Logic 게임에 적용하기 위한 방법을 제시하였다. 이를 구현하기 위해 제안된 방법은 임계치 결정방법, 이진 이미지에서 목표 이미지를 검출하여 게임에 적용하는 방법으로 대별할 수 있다.

스캔한 로고 이미지, 핸드폰 카메라 및 디지털 카메라로 촬영한 사진, 인터넷 웹페이지 캡처 이미지 등 총 10종의 이미지를 대상으로 실험한 결과 본 논문에서 제안한 임계치 결정 방법은 기존의 방법에서 검출이 곤란한 영역까지 검출 하여 게임 데이터에 적용할 수 있음을 알 수 있었다.

또한 제안한 목표 이미지 검출 방법 및 게임에 적용하는 방법은 임계치 결정 방법에 크게 영향을 받지 않고 지정된 가로 세로의 비율을 만족하는 크기로 검출되어 가로 세로 최대 픽셀에 대한 제한 사항을 준수하여 게임 데이터에 적용되었음을 확인하였다.

본문에서 제시하지 않은 특정 회사의 로고나

마크, 만화 캐릭터 등 6개의 이미지의 경우 임계치 결정방법에 따른 목표 이미지의 크기 차이는 있었으나 제시한 제한 조건을 준수하고 게임 테이터로 적용됨을 확인 하였다.

참 고 문 헌

[1] 권기현, “수토쿠 퍼즐을 통해서 살펴본 SAT에서 전처리 효과”, 한국IT서비스학회지, 제7권 제2호, pp. 127-135, 2008.

[2] 김기범 외, “3차원 최적 미로 게임 개발”, 한국해양정보통신학회 춘계종합학술대회, pp. 113-116, 2007.

[3] 정갑중, 이영준, “Java를 이용한 최적 미로 게임 개발”, 한국해양정보통신학회 춘계종합학술대회 논문집, 제9권 제1호, pp. 671-674, 2005.

[4] Liju Dong, Ge Yu, Philip Ogunbona, Wanqing Li, “An efficient iterative algorithm for image thresholding”, Pattern Recognition Letters, 29, pp. 1311-1316, 2008.

[5] 김광백, “밝기 정보를 이용한 영상 이진화에 관한 연구”, 한국해양정보통신학회 논문지, 제8권 제3호, pp. 721-726, 2004.

[6] 조규철, 마용범, 이종식, “GPCR 분류에서 ART1 군집화를 위한 퍼지기반 임계값 제어 기법”, 한국 컴퓨터 정보학회논문지, 제12권 제6호, pp. 167-175, 2007년 12월.

[7] 박영재, 장석우, 김계영, “눈 주위의 피부색을 이용한 피부영역검출과 입술검출에 관한 연구”, 한국 컴퓨터 정보학회논문지, 제14권 제4호, pp. 19-30, 2009년 4월.

[8] 장석우, 김상희, 김계영, “명암 필터와 개선된 허프 변환을 이용한 성인영상 검출”, 한국 컴퓨터정보학회논문지, 제14권 제5호, pp. 45-54, 2009년 5월.

[9] 손영선, “Japanese Puzzle을 이용한 교통안전 표지판 인식”, 한국지능시스템학회 논문지, 제18권, 제3호, p p. 416-421, 2008.

[10] 이대호, 박영태, “영상처리를 위한 C++ 프로그래밍”, 도서출판 인터비전, pp. 248-256, 2008년.

[11] P. K Sahoo, S. Soltani. and A. K. C. Wong, “A Survey of Thresholding Techniques”, Computer Vision, Graphics, And Image Processing, 41, pp. 233-260, 1988.

오 감 석



1993년 :부경대학교 대학원 (공학 석사)

1998년 :동경공업대학 대학원 (공학박사-지능시스템)

1998년~2005년 : 동명대학 조교수

2006년~현 재 : 동명대학교 정보통신공학과 조교수
관심분야 : 소프트웨어, 영상처리, 인공지능, 임베디드시스템