

움직임 검출과 영역 분할을 이용한 실시간 입체 영상 변환

권병헌* 서범석**

요약

본 논문에서는 2차원 정지 영상 및 동영상에서 블록 정합을 이용한 움직임 검출과 영역 분할을 통하여 생성한 깊이 지도를 이용하여 입체 영상으로 실시간 변환하는 방법을 제안하였다. 성능 평가는 움직임 객체의 깊이 지도와 절대 시차 차이 영상을 생성하여 기존의 변환 방법과 비교를 통해 제안한 방식의 우수성을 입증하였다.

Real-Time Stereoscopic Image Conversion Using Motion Detection and Region Segmentation

Byong-Heon Kwon*, Bum-Suk Seo**

Abstract

In this paper we propose real-time conversion methods that can convert into stereoscopic image using depth map that is formed by motion detection extracted from 2-D moving image and region segmentation separated from image. Depth map which represents depth information of image and the proposed absolute parallax image are used as the measure of qualitative evaluation. We have compared depth information, parallax processing, and segmentation between objects with different depth for proposed and conventional method. As a result, we have confirmed the proposed method can offer realistic stereoscopic effect regardless of direction and velocity of moving object for a moving image.

Key words : MTD, Matsumoto, Depth Map

1. 서론

1938년 처음으로 CRT(Cathode Ray Tube)를 이용한 TV 상용화를 시작으로 평면CRT, LCD, PDP, OLED에 이르기까지 디스플레이(Display) 장치의 개발은 빠르게 성장해 왔다. 이러한 디스플레이 기술의 발달은 정보 통신 분야에 눈부신 발전을 가져왔고, 보다 다양한 정보 즉, 영상과 음성 정보가 아닌 입체감과 현실감등의 느낌의 정보까지도 포함하는 입체 영상(Stereoscopic Image) 정보를 요구하게 되었다.

이러한 입체 영상 기술은 기존의 2차원 영상과는 달리 사람이 보고 느끼는 실세계의 영상과 거의 흡사하여 시각정보의 질적 수준을 한 차원 높여주는

새로운 개념의 영상 미디어로서 차세대 디지털 영상 문화를 주도하게 될 것으로 전망되고 있다.

기존 2차원 영상의 경우 양안이 동일한 영상을 보게 되므로 사람이 실제 사물을 보는 것과는 달리 불편함을 느낀다. 하지만 사람들은 지금까지 2차원 영상을 보아온 경험에 의해 불편함 없이 평면으로 보는 것이다. 사람들은 보다 더 실감나는 영상을 보기를 원하므로 2차원 영상에서 3차원 영상 제작에 많은 투자를 하고 있는 실정이다. 그 방식 중 하나는 두 대의 카메라를 통해 좌안 영상과 우안 영상을 각각 찍어서 만드는 방식과 기존의 2차원 영상을 수작업을 통해 한 장면 한 장면씩 3차원 영상으로 변환하는 방식이 있다.

* 제일저자(First Author) : 권병헌

접수일 : 2005년 7월 10일, 완료일 : 2005년 7월 26일

* 유학대학 정보통신공학과 부교수

** (주)플랫디스 책임연구원

2차원 영상을 3차원으로 일일이 수작업을 통해 변환하는 방식은 막대한 비용과 시간이 소요되는 단점이 있고 기존의 2차원으로 제시된 많은 양의 영상을 3차원으로 변환할 수 없는 단점이 있다.

이러한 단점들을 보완하고 효율적인 입체 시스템을 구현하기 위해 입체 영상 변환 기술이 90년대 초부터 개발되기 시작했다.

입체 영상은 그림 1과 같이 양안을 통해 보이는 서로 다른 영상을 인식함으로써 인간의 뇌에서 3차원 영상이 합성된다. 이러한 입체 영상을 재현하기 위해서는 왼쪽과 오른쪽 영상사이의 수평적 거리(차이)를 두어서 양쪽 영상의 깊이(Depth) 정보를 표현해야 한다[1].

본 논문에서는 차세대 디지털 영상 문화를 주도할 입체 영상의 원리와 입체 디스플레이 방식, 2차원 영상을 입체 영상으로 변환하는 기존의 방법에 대해 알아보고 기존 입체 변환 방법보다 효율적이고 입체 실감이 뛰어난 입체 변환 알고리즘을 제안한다.

제안한 알고리즘은 2차원 영상에서 움직임 검출과 영역 분할을 이용한 깊이지도(Depth Map)를 생성하고 양안 시차(Binocular Disparity)를 적용하여 서로 다른 원근 오행을 갖는 입체 영상을 생성하고, 기존의 기술과는 달리 2차원 영상에서 운동 물체의 운동 방향과 속도에 상관없이 입체 효과를 제공할 수 있다. 또한 실시간으로 입체 변환이 이루어짐으로 그 효율성이 높다.

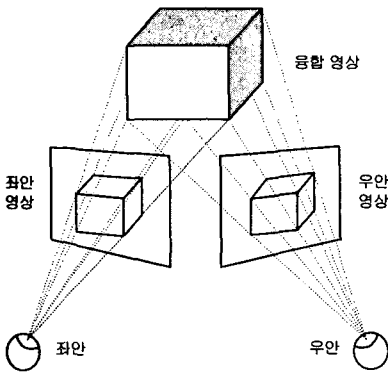


그림 1. 입체시의 원리

2. 기존의 입체 변환 방법

2.1 MTD(Modified Time Difference) 방식

MTD 방식은 (그림 2)와 같이 영상에서 물체가 우측으로 이동하고 있고, 카메라는 정지하고 있는 상태일 때, 현재 (N)번째 영상을 좌 영상으로 하고,

(N-2)번째 영상을 우 영상으로 구성하여 입체 영상을 만든 후 양안에 디스플레이하면 물체가 모니터 앞쪽으로 튀어나오듯이 보이므로 입체감을 느끼게 해준다[2][3]. 이 방식은 그림 2와 같이 움직이는 물체가 비교적 저속의 수평 운동일 때만 입체 효과가 좋고, 좌우 영상이 바뀌게 되면 물체는 배경 뒤에 있는 듯이 인식된다. 즉, 인간의 3차원 인식과 상반되는 현상이 발생하므로 눈의 피로감이 발생하게 된다. 그리고 움직이는 물체의 방향이 수평이 아닌 운동일 경우 움직이는 물체는 하나의 상으로 융합하지 못하고 2중상으로 보여 입체 효과를 얻을 수 없다.

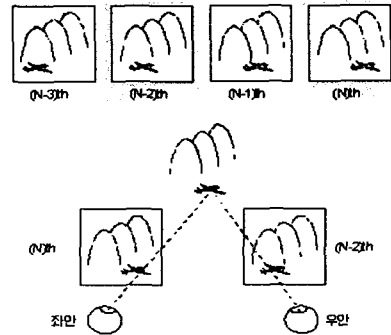


그림 2. MTD 방식의 원리

2.2 Matsumoto 방식

Matsumoto는 영상의 깊이 정보를 이용하여 스테레오(Stereo) 영상을 구현하였다. 이 기술은 산요 전기의 상업용 2차 제품에 적용되었는데, 저속 운동 영상인 경우에는 영상 운동을 추출한 후 운동 기반 깊이 결정 알고리즘을 이용하여 현재 영상 블록(Block)의 깊이 값을 추출한 후, 컴퓨터 그래픽에 사용하는 원근 투영을 거쳐 좌우 영상을 제작하였다[4]. 단점은 원근 투영 때문에 영상 왜곡이 발생하여 화질이 낮아지는 것이다. 고속 운동이 존재하는 영상보다 카메라 및 물체의 움직임이 작을 때 적용하면 입체 효과가 좋다.

Matsumoto 방식은 운동 벡터를 계산하기 위해 MPEG-2의 블록 매칭(Block Matching) 기술을 이용하였다. 블록 매칭 기법은 보다 작은 하드웨어 복잡도 때문에 현재 사용되고 움직임 예측 방법 중 가장 보편화된 기술이다. 이 기술의 기본 원리는 그림 3에 도시한 바와 같이 이전 프레임(N-1)에서 변위 픽셀 (i_1, j_1) 이 중심이 되는 $I_{block} \times J_{block}$ 크기를 갖는 블록이 현재 프레임(N)에서 픽셀 (i_2, j_2) 이 중심이 되고 같은 크기를 갖는 블록으로 매칭 되는지 탐색한다.

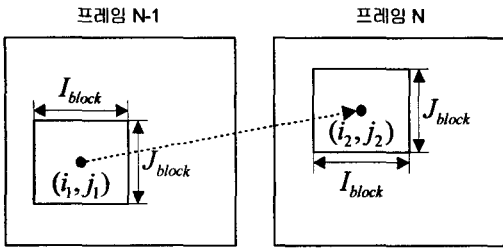


그림 3. 블록 매칭의 원리

여기서 매칭 척도로서는 블록 내에서 최소 MSE (Mean Square Error) 또는 MAD(Mean Absolute Difference) 값 등을 계산하여 매칭 여부를 정하며, 블록의 탐색 방법으로 3단계 탐색, 크로스 탐색 등이 있고, 블록 크기의 결정은 계층적, 적응적 방법 등 블록 매칭에 대한 다양한 기법들이 제안되어 있다[5][6].

3. 제안한 입체 변환 알고리즘

3.1 영역 분할 및 움직임 검출

제안한 움직임 검출 방식은 현재 및 이전 영상에 대한 표본 휘도 영상에서 정지 및 움직임 픽셀을 검출한다. 움직임 검출 방법은 임계값 기반의 윤곽 검출 및 미디언 필터링, 그리고 이들의 AND 연산으로 구성된다. 제안한 방식에서 미디언 필터링은 움직임 검출의 정확도를 높이고, 임계값 기반의 윤곽 검출은 움직임 픽셀 중 배경을 검출하는 오류를 최소화한다.

먼저 윤곽 검출은 동차 연산자(Homogeneity Operator)를 이용한다. 이 방법은 윤곽 검출을 가장 빠르고, 간단하게 할 수 있다[7]. 그림 4와 식(1)에 도시한 바와 같이 현재 표본 휘도 영상에서 3x3 서브 윈도우를 취하고, 그 윈도우의 중심 픽셀을 중심으로 인접한 8개의 픽셀과 절대 차이를 구하여, 이들 중 최대값을 그 서브 윈도우의 윤곽으로 정한다.

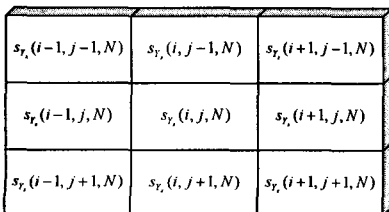


그림 4. 현재 프레임의 3x3 서브 윈도우

$$s_{Y_{EDGE}}(i, j) = \max\{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\} \quad (1)$$

일반적으로 미디언 필터링은 신호 처리에서 사용되는 비선형 기술 중 가장 대표적인 것으로서, 영상에 포함된 객체를 검출하기 위해 형태 정보를 이용한다. 특히, 임펄스 성분이 포함된 노이즈 패턴에 효과적이다[7]. 본 논문에서는 현재 및 이전 표본 휘도 영상의 픽셀의 절대 차이에 대한 형태 정보를 이용하기 위하여 그림 5와 같이 3x3 윈도우에서 미디언 필터링을 행한다.

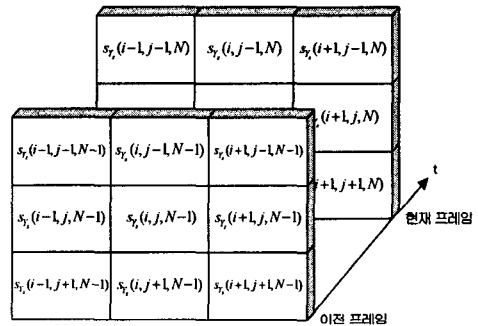


그림 5. 현재 및 이전 프레임의 서브 윈도우

$$s_{Y_{MED}}(i, j) = MED\{m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9\} \quad (2)$$

상기한 미디언 필터링 결과를 임계값을 이용해서 영상 내 객체에 대한 형태 정보의 유무를 추출한다. 여기서, 형태 정보가 있을 때는 255, 없을 때는 0으로 정한다. th_{MED} 는 미디언 필터링 임계값으로 영상 내 존재하는 노이즈 마진을 고려한 것이다.

$$s_{Y_{OR}}(i, j) = \begin{cases} 0, & \text{for } s_{Y_{MED}}(i, j) \leq th_{MED} \\ 255, & \text{for } s_{Y_{MED}}(i, j) > th_{MED} \end{cases} \quad (3)$$

상기한 객체에 대한 형태 정보의 정확도를 높이기 위하여 윤곽 검출값과 비트별 AND 연산을 취하고, 임계값 비교를 통해 최종적으로 정지(0) 및 움직임(1) 픽셀을 검출한다.

움직임 검출의 오류를 보완하기 위해 본 논문에서는 블록 정합 기법 중 고속 탐색 방법을 이용하여 움직임 벡터를 추출한 후 윤곽선 검출과 영역 분할을 통해 산출된 움직임 픽셀과 움직임 벡터 크기의 유사도를 계산하여 최종 깊이 정보를 생성하였다.

움직임 벡터를 추출하기 위해 고속 탐색 기법 중

계층적 탐색 방법을 이용하였다. 계층적 탐색 방법은 상위 단계로 올라갈수록 공간 해상도가 낮아지는 평균 피라미드(Mean Pyramid)를 생성하고 상위 단계에서 찾은 움직임 벡터를 중심으로 작은 범위의 움직임 추정을 하위 단계에서 수행하여 하위 단계의 움직임 벡터를 찾는다[6].

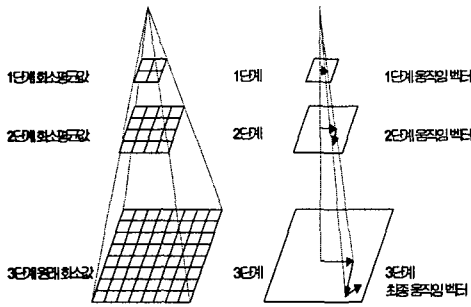


그림 6. 블록 매칭 방법 - 계층적 탐색

전체 움직임 탐색 영역은 16x16이며 평균 절대 오차(MAE: Mean Absolute Error)를 정합 기준으로 사용하였다.

$$MAE(i, j) = \frac{1}{MN} \sum_{(x,y) \in B} |X(x, y, t) - \hat{X}(x-i, y-j, t - \Delta t)| \quad (4)$$

M, N은 서브 윈도우의 해상도를 나타내고 B는 움직임 탐색 영역을 나타낸다. 본 논문에서는 3x3 서브 윈도우를 설정하고 현재 및 이전 화면에서 블록 정합을 통해 1단계부터 3단계까지의 움직임 벡터의 크기를 검출했다. 움직임 벡터의 크기(MV)에 따라 깊이 정보(D_{MV})를 설정하여 움직임 객체와 배경을 구분하였다.

3.2 오행 지도(Depth Map) 생성

표본 휘도 영상에서 각 영역 당 계산된 8개의 평균과 표준 편차를 이용하여 원래 영상의 해상도 크기의 깊이 지도를 작성한다. 즉, 원래 영상의 휘도값이 실험 결과에 의하여 $\mu_R \pm \sigma_R$ 의 범위에 있으면 원래 영상의 픽셀을 움직이는 객체를 구성하는 픽셀로 정하고, 상대적으로 가까운 위치에 있는 영역이므로 깊이 정보를 0으로 설정해주며, 그 외의 픽셀은 깊이 정보를 1로 설정한다.

$$s_D(i, j) = \begin{cases} 0, & \text{for } \mu_R - \sigma_R \leq s_Y(i, j) \leq \mu_R + \sigma_R \\ 1, & \text{otherwise} \end{cases} \quad (5)$$

여기서, s_D(i, j)는 원래 해상도를 갖는 깊이 정보이다. 깊이 정보(S_D)를 이용해서 윤곽선 검출 기반의 깊이지도 영상(D_S)을 구한다.

$$D_S = \begin{cases} 0, & \text{for } S_D = 0 \\ 255, & \text{for } S_D = 1 \end{cases} \quad (6)$$

윤곽선 검출 기반의 깊이지도 영상(D_S)과 움직임 추정 기반의 깊이지도 영상(D_{MV})의 차 영상(D_S - D_{MV})을 계산하여 움직임 크기가 클수록 어둡게 표현되고 움직임 크기가 작을수록 밝게 표현되는 최종 깊이지도 영상을 생성한다. 이는 국부극소에 의한 성능 저하가 발생하는 계층적 탐색 방법의 단점을 보완하고 기존 방법에 비해 정확한 깊이 정보를 생성할 수 있다.

3.3 시차(Parallax) 처리

앞에서 생성한 오행 지도(Depth Map)에 따라 영상의 각 픽셀에 대해 시차 처리를 하게 된다. 상대적으로 가까운 물체는 음 시차(Negative Parallax) 처리를 하여 물체의 상이 스크린 앞에 튀어나와 보이게 하고, 상대적으로 먼 물체는 양 시차(Positive Parallax) 처리를 하여 스크린 안쪽에 위치하도록 해야 한다.

그림 7은 오행 지도에 따라 좌안 영상, 우안 영상에 대해 시차 처리 방법을 나타낸다. 좌안 영상일 경우 깊이값이 0일 때 우로 지연시키고 우안 영상일 때는 깊이값이 1일 때 우로 지연 시킨다. 즉, 깊이값이 0인 픽셀군은 음 시차(Negative Parallax)가 적용되고 깊이값이 1인 픽셀군은 양 시차(Positive Parallax)가 적용되는 것이다[8].

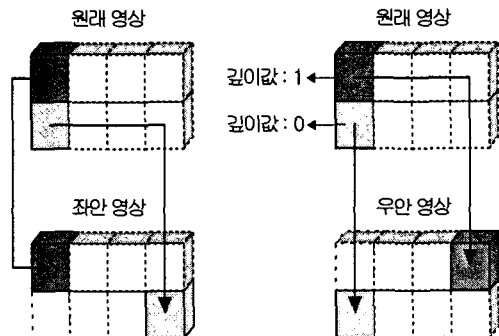


그림 7. 시차 처리

4. 실험 결과

4.1 실험

본 실험에서는 동영상 내 객체의 움직임 방향과 속도의 영향에 따른 성능 비교를 수행하기 위해 실험 영상을 수평 운동 영상과 수직 운동 영상으로 구분하였고 동영상 프레임율을 조절하였다.

성능 평가를 효율적으로 하기 위해 기존 입체 영상 변환 방법인 MTD 방식과 Matsumoto 방식의 깊이 정보와 시차 차이 영상과 제안한 방식에 의해 생성된 깊이 정보와 시차 차이 영상을 비교해서 배경 및 운동 객체의 오행을 적절히 적용했는지를 판단하고, 물체의 윤곽 여부를 조사하여 오행 처리 효과를 비교하였다.

4.2 결과 분석

Matsumoto 방식의 경우 블록 매칭 기법에 사용된 블록 크기는 4×4 이며, 탐색 영역인 매크로 블록 크기는 16×16 이다. 이 방식의 깊이 지도를 살펴보면, 꽃 영상인 경우 중앙의 나무와 나무 뒤의 정원과 집이 움직임 객체로 검출되었다. 하지만, 객체가 4×4 블록 크기로 표현되어 객체와 배경 사이의 윤곽 표현이 제대로 이루어지지 않았으며, 집 앞의 작은 나무와 같은 미세한 객체에 대해서도 동일한 결과를 확인할 수 있다.

반면 제안한 방식의 깊이 지도를 살펴보면, 꽃 영상인 경우 나무와 정원 그리고 집이 모두 움직임 객체로서 윤곽이 뚜렷하게 검출된 것을 확인할 수 있으며, 깊이 단계를 2단계로 설정했기 때문에 움직임 객체간의 깊이 정보는 동일하다는 것을 알 수 있다.



그림 8. Matsumoto 방식의 오행지도



그림 9. 제안한 방식의 오행지도

꽃 영상은 왼쪽에서 오른쪽으로 수평 이동하는 영상으로 제안한 방식은 움직임 객체의 윤곽이 뚜렷하게 나타난다. 하지만 Matsumoto 방식의 경우 다른 방식에 비해 나무나 집의 윤곽이 모호하게 나타난 것을 확인할 수 있다. 이는 깊이 정보가 정확하지 않기 때문이다.

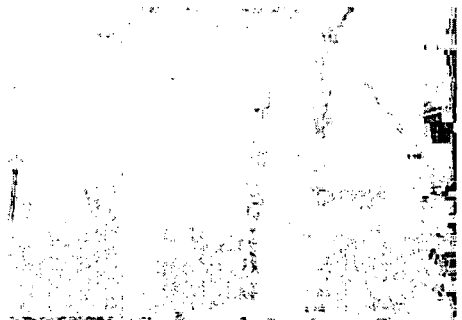


그림 10. Matsumoto 방식의 시차 차이 영상

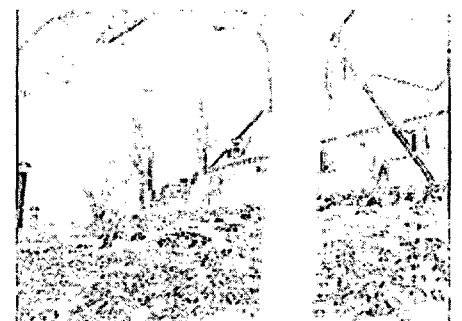


그림 11. 제안한 방식의 시차 차이 영상

5. 결론

본 논문에서는 2차원 영상의 움직임 검출과 영역 분할을 이용해서 깊이 정보를 산출하고, 그 정보에 따라 영상의 각 픽셀에 대해 음 시차와 양 시차 처

리를 수행함으로써 입체 영상으로 변환하는 알고리즘을 제안했다.

제안한 입체 변환 방법은 2차원 영상을 실시간으로 입체 변환이 가능하고 운동 시차를 매개 변수로 깊이 정보를 추출하기 때문에 기존 방식에 비해 객체의 운동 방향이나 속도의 영향에 의한 성능 저하를 줄일 수 있었다.

2차원 영상을 실시간으로 입체 변환이 가능함으로 3차원 영상 제작에 소요되는 시간과 비용을 절감할 수 있고 기존의 다양한 2차원 영상(TV, DVD, VTR)을 입체로 볼 수 있다는 장점이 있다.

제안한 방식은 100,000 게이트 소요로 하드웨어 구현이 가능하며, 현재 상용 중인 시분할 방식, 편광 필터 방식, 패럴랙스 배리어 방식, 렌티큘러 방식과도 호환이 가능하여 다양한 입체 변환 디스플레이 시스템을 구축할 수 있다.

참 고 문 헌

- [1] S. Nagada, *How to Reinforce Perception of Depth in Single Two-Dimensional Pictures*, Tyler & Francis, 1991.
- [2] Lipton, et al., "Stereoscopic Television System with Field Storage for Sequential Display of Right and Left Images," U.S. Patent No. 4,562,463, 1985.
- [3] H. Murata, et al., "Conversion of Two-Dimensional Image to Three Dimension," SID95 DIGEST, pp. 859-862, 1995.
- [4] Y. Matsumoto, et al., "Conversion System of Monocular Image Sequence to Stereo using Motion Parallax," SPIE Photonic West, vol. 3012, pp. 108-115, 1997.
- [5] H. Gharavi, et al., "Block-Matching Motion Estimation Algorithms : New Results," IEEE Trans. Circ. and Syst., Vol. 37, pp. 649-651, 1990.
- [6] M. Bierling, "Displacement Estimation by Hierarchical Block-Matching," Proc. Visual Comm. and Image Proc., SPIE Vol. 1001, pp. 942-951, 1988.
- [7] Randy Crane, *A Simplified Approach to Image Processing - Classical and Modern Techniques in C*, Prentice Hall, 1997.
- [8] Douglas J. Smith, *HDL Chip Design - A Practical Guide for Designing, Synthesizing and Simulating ASICs and FPGAs using VHDL or Verilog*, Doone Publications, 1996.

권 병 헌



1987년 한국항공대학교 항공전자공학과 공학사
 1989년 한국항공대학교 대학원 항공전자공학과 공학석사
 1995년 한국항공대학교 대학원 전자공학과 공학박사
 1997년 (주)LG전자 선임연구원
 1997 ~ 현재 유학대학 정보통신학과 부교수
 관심분야 : 영상신호처리, 통신, 3D displays.

서 범 석



1999년 한국항공대학교 항공전자공학과 공학사
 2001년 한국항공대학교 대학원 항공전자공학과 공학석사
 2003년 한국항공대학교 대학원 항공전자공학과 박사과정 수료
 2003년 ~ 현재 : (주)플렛디스 책임연구원
 관심분야 : 영상신호처리, HDL을 이용한 하드웨어 디자인