

□ 기술애설 □

가상공간 탐색을 위한 3D 인터액션 기법들[†]

숭실대학교 정문렬*

1. 서 론

3차원 가상공간은 이제 일상적인 것이 되어 가고 있으며 인터넷 상에서도 3차원 가상공간을 접할 수 있다. 가상공간(Virtual Space)을 구축을 위한 VRML[1](Virtual Reality Modeling Language)이 표준으로 자리잡은 이후 많은 홈페이지가 VRML을 이용하여 구축되었고 사용자와 가상공간과의 상호작용이나 행위 기술기능이 추가된 VRML 2.0 규약이 발표됨에 따라 그 이용은 더욱 증가하고 있는 추세이다[2]. 현재 다수의 VRML 브라우저가 개발되어 사용되고 있다. 이들 브라우저는 나름대로의 탐색 인터페이스(가상카메라 제어방법)를 가지고 있는데, 이것이 얼마나 만족스러운지 살펴보는 것도 흥미로운 일이다. 이를 위해 본 글에서는 먼저 많이 사용되고 있는 브라우저들의 가상공간 탐색 인터페이스를 분석하고 이들 탐색 인터페이스의 기본적인 접근방법 및 문제점들을 파악한다. 그 후 지금까지 여러 가지 목적으로 제시된 3차원 물체 조작 및 공간탐색 기법들을 살펴보고 이들을 좋은 가상공간 탐색 기법을 구현하는데 어떻게 활용할 수 있을 것인지 살펴본다. 공간탐색 기법들중 사용자가 가상공간내의 목표위치를 지정하고 사용자가 그 목표위치 가까이 가도록 카메라 피라미터(위치 및 방향)를 자동적으로 계산해주는 기법[10]과, 물체조작이나 다른 아바타와 상호작용시 그 상황을 가장 잘 보여주도록 카메라의 피

라미터를 자동적으로 계산해주는 기법[11]은 공간탐색의 범주에 들지 않는다고 보고 본 글의 검토에서는 제외하였다. 본 글은 사용자와 3차원 공간간의 상호작용에 관심이 있는 사람들과 앞으로 가상공간 브라우저를 개발하고자 하는 사람들에게 도움이 되리라 생각된다.

2. 기존 브라우저 분석

기존 브라우저들의 탐색 인터페이스가 가지고 있는 문제점들을 파악하기 위해 우선 대표적 브라우저인 코스모플레이어(Cosmo Player)와 Live3D의 탐색 인터페이스 기능에 대하여 간단히 살펴본다. 이들 브라우저는 3차원 공간을 사용자에게 보여줄 때 컴퓨터 스크린에 3차원 공간을 투영시킨다. 그리고 사용자로부터 입력을 받아들일 때 보통의 2차원 마우스를 사용한다.

코스모플레이어(Cosmo Player)[3] 두 가지의 탐색 모드, 즉 걷기(walk) 모드와 조사(examine) 모드를 제공하고 있다. Walk 모드를

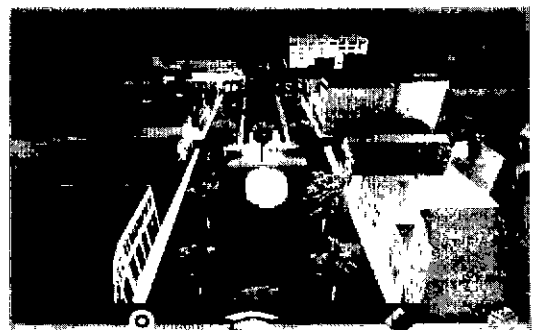


그림 1 코스모플레이어(Cosmo Player)의 Walk모드

*중심회원

† 본 원고는 시스템공학연구소의 연구과제(과제번호 9P00200) 수행의 일환으로 쓰여짐

선택하면 그림 1의 하단에 배치되어 있는 세 가지 조종간(부모드)이 나타나고 사용자는 이를 이용하여 탐색한다. 제일 왼쪽에 있는 조종간(look around 모드)을 마우스로 선택한 상태에서 마우스를 전후좌우로 이동시키면 가상공간을 보여주는 카메라는 상하좌우로 회전하고, 가운데 있는 조종간(walk around 모드)을 마우스로 선택한 상태에서 마우스를 전후로 이동시키면 카메라는 전후로 이동하고 마우스를 좌우로 이동시키면 카메라는 좌우로 회전하며, 오른쪽에 있는 마름모꼴의 조종간(Panning 모드)을 선택한 상태에서 마우스를 상하좌우로 이동시키면 카메라는 상하좌우로 이동한다. 조사모드는 어떠한 대상을 관찰하고자 할 때 사용하는데, 이 모드를 선택하면 그림 2의 하단에 배치되어 있는 세 가지 조종간이 나타난다. 제일 왼쪽에 있는 조종간(Dolly In/Out 모드)을 선택한 상태에서 마우스를 전후로 이동시키면 카메라는 전후로 이동하고, 가운데에 있는 구(Rotate 모드)를 선택한 상태에서 마우스를 전후좌우로 이동시키면 카메라는 관찰하고자 하는 대상에서 일정한 거리만큼 떨어진 구의 표면에 하상우좌로 움직이고, 오른쪽에 있는 마름모꼴 상자(Panning 모드)를 선택한 상태에서 마우스를 전후좌우로 이동시키면 카메라의 위치는 상하좌우로 이동한다.

Live3D는 walk, look, slide, point, fly 모드를 제공한다. Walk모드를 선택하고 마우스를 전, 후, 좌, 우로 이동시키면 카메라를 전후로 이동하거나 좌우로 회전시킬 수 있고, Look모드는 카메라의 위치는 고정된 상태에서 방향만 회전시키며, Slide모드는 카메라의 위치를 상하

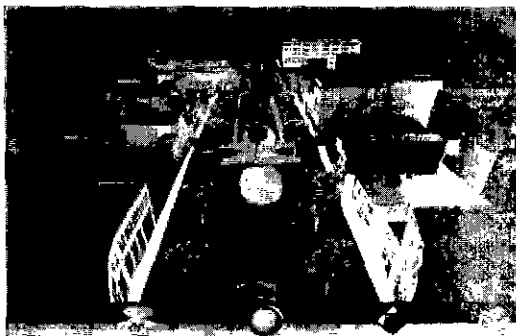


그림 2 코스모플레이어(Cosmo Player)의 Examiner모드

좌우로 이동시키면서 전에 바라보던 곳을 계속 관찰하게 해주며, Point모드는 마우스로 클릭한 위치를 관찰할 수 있도록 카메라의 위치와 방향을 이동시켜준다. fly 모드를 선택하면, 비행기가 나는 것과 같이 오른쪽이나 왼쪽으로 기울어 날아가는 것처럼 카메라를 이동시킬 수 있다.

앞에서 언급한 대표적인 2개의 브라우저뿐만 아니라 그 외의 다수의 브라우저를 분석해보면 이들이 다음과 같은 문제점들을 가지고 있음을 알 수 있다.

대부분의 브라우저는 여러 가지 탐색 모드를 제공하고 있는데, 적게는 2개의 모드부터 많게는 7개까지의 모드를 제공하고 있다. 그러나 모드를 분류하는 기준이 분명하지 않아서 언제 어떤 모드를 선택해야할지 불분명하다. 한편 각 모드 내에서 사용자는 조종간을 조작하도록 되어 있는데 이를 어떻게 조작해야 하는지 그것이 어떤 기능을 하는지 기억하기가 쉽지 않다. 또한 모드간에 동일기능이 중복되어 있다. 예를 들면, 코스모 플레이어의 경우, 조사모드에서 카메라를 전후로 이동시킬 수 있는데(Dolly In/Out 조종간), 이는 견기모드에서의 walk around 조종간에서도 가능하다. 그리고 조사모드와 견기모드 모두에서 Panning(카메라의 좌우상하운동)이 가능하다.

이것은 모드를 바꾸지 않고 현재모드에서 가능하면 많은 일을 하기 위함이다. 그러나 모드간 기능의 중복은 사용자를 혼돈시키는 경향이 있다. 그리고 특정 모드, 특정 조종간에서 카메라를 조작할 때, 마우스의 움직임과 카메라의 움직임의 대응관계가 일관성을 결여하고 있는 경우도 있다.

예를 들어, 코스모 플레이어의 경우, 견기(walk) 모드의 walk around 조종간에서, 마우스의 전후이동은 카메라의 전후이동으로 해석되고 마우스의 좌우이동은 카메라의 좌우회전으로 해석된다.

이러한 설계는 견기에서 제일 중요한 일은 전후로 이동하는 것과 좌우로 고개를 돌려 주위를 관찰하는 것이라는 인식에 그 근거를 두고 있다. 그러나, 마우스는 이동운동을 하는데 카메라는 회전운동을 하게 하는 것은 입력 장

치의 운동과 카메라의 운동간의 대응성을 파괴하기 때문에 문제가 있다.

3. 3D 인터액션 파라다임

지금까지 VRML브라우저들의 탐색인터페이스 기법을 살펴보았다. 이 문제는 기본적으로 3차원 가상공간에 있는 자유도 6을 가지는 가상카메라를 어떻게 제어하느냐는 문제이다. 문제를 일반 물체에까지 확대시키면 이는 3D 인터액션(3D interaction) 문제가 된다. 3D 인터액션 문제의 해결을 기법들은 크게 세가지 파라다임으로 나누어 생각할 수 있다. 각 파라다임을 소개하고 이 파라다임이 가상공간 탐색에 적절한지 그리고 각 파라다임내의 어떤 기법이 가상공간 탐색에 적절한지 살펴본다.

역사적으로 가장 최근의 파라다임은 가상현실이라 할 수 있다. 현재 가상현실기법은 전자장갑이나 스페이스 볼 등의 3차원 입력장치를 사용하여 사용자로부터 입력을 받고 사용자가 가상공간에 몰입될 수 있도록 스테레오 안경이나 헤드 마운트 디스플레이 등을 이용해 3차원 공간을 제시한다. 가상현실기법에서는 전자장갑을 이용하여 손 제스처를 입력받은 후 이 제스처에 따라 가상카메라를 제어할 수 있다. 그러나 손 제스처를 사용해서 원하는 위치까지 도달하도록 카메라를 제어하는 문제가 생각만큼 쉽지 않다. 이것은 영화적일때 전문 카메라 기사가 필요한 이유를 생각해 보면 알 수 있다. 보다 전통적인 파라다임은 입력장치로는 3차원 또는 이에 준하는 장치를 쓰되 3차원 공간을 컴퓨터 스크린밖에서 스크린에 투영된 형태로 바라보게 하는 것이다. 이를 "윈도우를 통하여" 파라다임이라 부를 수 있다[8]. 3D 마우스를 이용한 예는 SimGraphics Engineering Corp.에서 만든 Flying Mouse를 들 수 있다[7]. Flying Mouse는 가상공간 탐색보다는 물건을 선택하고 조작하는 일에 주로 쓰였는데, 항상 마우스를 공중에 든채 버튼을 눌러야 한다는 것이 적지 않게 불편하다. 물론 우리가 일상생활에서 TV나 비디오를 사용할 때 리모트 콘트롤러를 든채 버튼을 누르지만, 이 경우는 일시적으로만 그렇게 하기 때문에 경우가

다르다[8]. 이런 점을 고려해 [8]에서는 수직 방향의 자유도는 없이 자유도 5를 가지는 "Desktop Bat"를 제안하였다. Desktop Bat는 평평한 바닥위에 반구(hemisphere)가 놓여 있는 모양을 하고 있다. 바닥은 평면상에서 움직이고, 반구는 트랙볼처럼 임의의 축을 중심으로 자유로이 회전할 수 있도록 되어 있다. Desktop Bat는 기본적으로 일반 마우스에 pitch, roll, yaw 회전(또한 이들의 조합)을 추가한 것이다. Desktop Bat의 움직임은 공간내에서 3차원 커서(가상 Bat)로 표시된다. Desktop Bat의 반구를 회전시키면 가상 Bat의 바닥은 그 회전을 표시하기 위하여 방향을 바꾼다. Desktop Bat의 바닥을 평면상에서 움직일 때, 가상 Bat는 현재의 바닥평면상에서 움직인다. 이것을 이용하면 Desktop Bat이 가지고 있지 않은 상하운동 자유도를 구현할 수 있다. 가상 Bat의 바닥이 수직평면과 일치되도록 Desktop Bat를 전후로 회전시킨후 Desktop Bat를 평면상에서 움직이면 가상 Bat는 수직평면상에서 움직인다. Bat는 5개의 버튼을 가지고 있는데 이들은 다섯 손가락으로 각각 누를 수 있는 위치에 배열되어 있다. 버튼을 조합하여 누르면 대응되는 행위를 발생시킨다. 예를 들어, 엄지버튼과 집게버튼을 동시에 누르면 Bat의 현 위치를 나타내고 있는 밑에 있는 물체가 선택이 되어, 버튼을 누르고 있는 동안 가상 Bat에 부착되어 이와 같이 움직인다. 또한 가상카메라를 가상 Bat의 위치/방향으로 옮겨 보이는 장면을 바꾸려면 엄지와 세끼 버튼을 누르도록 되어 있다. 따라서 엄지와 세끼 버튼을 누른채 Bat를 움직이면 가상카메라도 따라 움직인다.

Desktop Bat은 일반 물체를 선택하고 조작하는 데는 매우 유용한 도구로 사용될 수 있다고 생각된다. 그러나 이 Bat로 가상카메라의 방향을 제어하는 일은 쉽지 않다. Bat의 회전은 트랙볼의 회전과 같이 회전축이 고정되어 있지 않고 임의의 축을 중심으로 회전할 수 있기 때문에 특정방향을 중심으로 회전하는 일이 매우 어렵다. 가상카메라는 많은 경우 특정축을 중심으로 흔들림이 없이 회전시키는 것이 요구된다. 그렇지 않으면 장면이 심하게 흔들

리는 결과를 야기할 수 있다. 그러나 Desktop Bat은 충분히 고려해 볼 수 있는 도구이다.

3D 상호작용의 가장 전통적인 파라다임은 적절한 상호작용 모델을 이용하여 2D 마우스 운동을 3D 조작명령으로 바꾸는 것이다[5, 6]. 여기서 소개되는 대부분의 방법은 3차원 공간에 있는 물체를 조작하기 위한 기법이지만 가상공간 탐색에 사용되는 가상카메라의 제어에 사용될 수 있다. 비록 가상카메라는 눈에 안보이지만 눈에 보이는 장면과 가상카메라의 위치/방향은 밀접한 관계가 있기 때문에 요구되는 장면의 변화에 대응되는 가상카메라의 운동을 어렵지 않게 예상할 수 있다.

[6]에서는 3차원 이동방법을 제시하고 있다. 이 방법은 마우스의 운동과 세계좌표계의 세축 벡터의 스크린상의 투영을 비교하여, 스크린 투영이 마우스의 이동방향과 가장 가까운 좌표축 방향으로 물체를 이동시킨다. 예를 들어, 스크린에서 보기에 세계좌표계의 x축이 왼쪽을 가리키고 있을 경우, 마우스를 왼쪽으로 움직이면 물체는 세계좌표계의 x축을 따라 움직인다. 그러나 이 방법은 두축이 스크린상에 서로 가까이 투영되면 어떤 축으로 물체를 움직여야 될지 불분명하다. [5]에서는 물체 회전방법을 제시하고 있다. 여기서 물체의 회전은 시선 방향을 축을 중심으로 하거나 시선방향에 수직인 축을 중심으로 한다. [5]에서는 이외에도 가상구(virtual sphere)라고 불리는 기법을 기술하고 있는데, 이는 물체의 원점에 위치한 트랙볼(trackball)을 시뮬레이트한 것이다. 사용자는 마우스로 트랙볼을 잡고 손가락 하나로 트랙볼을 돌리듯이 가상구를 전후좌우 또는 임의의 방향으로 돌린다. 이때 순간적으로는 회전축이 존재하지만 고정되어 있는 회전축은 없다. 이 기법은 물체를 이리저리 회전시켜 특정 방향에 일치시키는데는 좋으나 가상카메라가 가상구내에 들어있다고 생각하면 이 가상구 기법은 카메라 제어에도 쓰일 수 있다. 그러나 앞서 Desktop Bat의 경우에서도 언급했듯이 특정 회전축을 중심으로 카메라를 회전하는 것이 어려우므로 가상카메라의 제어에 가상구를 사용하는 것은 불편한 점이 있다.

카메라의 파라미터(위치와 방향)를 사용자가

직접 제어하지 않고 간접적으로 제어하는 기법도 제안되었다. 즉 사용자는 관찰하고 싶은 장면이나 방식을 적절한 방법으로 표시하고 시스템은 이를 가능하게 하는 카메라 파라미터를 자동적으로 구하는 기법이다. 대표적인 예로 "Through-the-Lens Camera Control"[9]을 들 수 있다. 이 기법에서는 사용자가 마우스를 이용해서 스크린상의 특정 점을 다른 점으로 옮기면 시스템이 이를 가능하도록 하는 카메라의 파라미터를 결정한다. 즉 공간상의 한 점이 스크린상에 투영되어 있는 경우 이 점이 스크린상의 다른 점에 투영되도록 카메라의 파라미터를 결정한다. 스크린상의 한 점뿐만 아니라 여러 점을 옮길 수도 있다. Through-the-Lens 제어기법은 가상 카메라의 방향을 제어하는데 유용한 기법이다. 즉 마우스의 한 버튼을 누른 채 상하좌우로 움직이는 것을 스크린상의 커서 밑에 있는 점을 상하좌우로 움직이는 것으로 보면 가상카메라의 방향을 제어하는 문제는 해결된다. 카메라의 위치조작은 마우스의 버튼을 누르지 않은 상태에서 상하좌우로 움직이는 것을 카메라를 가상공간내에서 전후좌우로 움직이는 것으로 보면 해결된다. 보다 확실한 결론을 얻기 위해서는 실험적 조사연구를 통하여 이 방법의 효율성을 확인할 필요가 있지만, Through-the-Lens 제어방식이 지금까지 검토한 방식 중에서 가장 직관적이고 사용하기 편해보이는 방식이다. 그러나 이 방법으로는 카메라를 가상공간내에서 상하로 움직이는 것을 구현할 수 없다. 물론 컨트롤 키를 사용해서 마우스의 전후운동을 카메라의 상하운동으로 간주하면 이 문제를 해결할 수 있지만 이런 식의 해결은 입력장치의 운동과 카메라의 운동간의 대응관계를 파괴하므로 바람직하지 못하다. 따라서 마우스의 자유도가 카메라의 자유도에 미치지 못하는 데 따른 문제를 보다 일관성있게 해결하는 방안을 생각할 필요가 있다.

마우스는 자유도가 2밖에 되지 않기 때문에 자유도가 6인 카메라를 조작하기 위해서는 카메라의 운동을 여러 가지 모드를 나누고 이를 조합하여 카메라를 조작할 수밖에 없다. 문제는 모드를 어떻게 나누고 어떻게 선택하는가 하는 것이다. 앞서 살펴 본대로 기존의 가상공

간 브라우저에서는 이 문제를 만족스럽게 해결하지 못하고 있다. 이 문제를 해결하는 한 접근방법으로 가상구(virtual sphere) 기법을 확대 적용하는 것을 생각할 수 있다. 가상구는 카메라의 운동공간의 부분공간을 제어할 수 있는 가상입력장치라고 볼 수 있다. 그 부분공간에서의 카메라의 운동은 2차원 마우스의 운동으로 전부 표현된다. 2차원 마우스의 운동으로 제어가 가능한 부분운동공간을 "2차원 부분운동공간"이라 하자. 이때, 카메라 제어문제는 카메라의 6차원 운동공간을 전부 포괄할 수 있는 2차원 부분운동공간들을 정의하고, 2차원 부분공간들을 제어하는 가상콘트롤러들을 스크린에 표현하고 이를 마우스로 제어하는 방법을 결정하는 문제로 귀결된다. 이 방법을 구현한 예로는 [4]를 들 수 있다. 이 기법에서는 카메라의 운동공간을 6개의 2차원 부분공간, 즉 평면상의 이동, 수직면상의 이동, 현위치중심의 피치(pitch), 요(yaw) 회전, 현위치중심의 롤(roll), 피치회전, 좌우상하방향의 선회, 목표물체중심 가상구면상의 회전 공간들로 분할하였다. 각 부분공간에서 가능한 카메라 운동을 직관적으로 나타내기 위해 적절한 보조면, 보조선, 보조원을 사용하였다. 예를 들어, 수직면상의 이동 공간은 수직 격자면상을 헬리콥터가 기어오르는 모습을 가지고 있다. 여기서 헬리콥터는 카메라를 상징하는데, 카메라가 공간상에서 움직인다고 생각하기 보다는 관찰자가 헬리콥터를 타고 움직인다고 보는 것이 일반 사용자에게 보다 직관적이기 때문에 헬리콥터 제어 비유를 사용하였다. 그러나 이 비유는 꼭 필요한 것은 아니라 헬리콥터대신에 카메라 모양을 가진 물체를 사용할 수 있다. 사용자는 원하는 관점을 지정하기 위하여 스크린에 표시된 6개의 부분운동공간을 마우스를 옮겨가며 선택하고 선택된 공간 내에서의 운동은 마우스의 전후좌우운동으로 제어한다. 이 기법을 "가상콘트롤러 기법"이라 하자. 이 기법의 문제점으로는 한 부분공간에서 다른 부분공간으로의 전환이 연속적이지 못하다는 것을 들 수 있다. 그러나 보통 한 장면의 관찰은 한 부분공간만을 사용하는 경우가 많으므로 이것이 크게 문제가 된다고 볼 수 없다. 카메라의 위치를 움직이면서

패닝(panning)하는 조작은 이를 두 부분공간을 동시에 제어해야 되기 때문에 할 수 없다. 그러나 보통 카메라를 좌우로 움직이면서 패닝하는 경우가 많으므로 이 경우를 2차원 부분운동공간으로 표시하면 이 문제를 해결할 수 있다. 그리고 일반적으로 카메라의 오일러 각(피치, 요, 롤)을 조절하면서 사용자가 원하는 카메라의 방향을 얻기가 매우 어렵기 때문에 가상콘트롤러 기법내의 카메라 회전공간 제어는 Through-the-Lens 기법으로 바꾸는 것도 좋은 방법이라 생각된다.

4. 결 론

이상 가상공간 탐색을 위한 카메라 제어 기법을 3D 인터랙션의 세가지 파라다임의 관점에서 살펴보았다. 각 파라다임은 응용분야에 따라 나름대로의 장단점이 있다. 본 글에서 주로 염두에 두었던 응용분야, 즉 인터넷상에서의 3D 가상공간 탐색을 위해서는 2D 마우스를 사용하는 가장 전통적인 파라다임으로 충분하고 또 사용하기에 편하다고 생각된다. 2D 마우스를 이용해서 가상공간을 탐색하기 위해서는 가상콘트롤러 기법과 Through-the-Lens 기법을 혼합하는 것을 시도해 볼만하다고 생각된다.

참고문헌

- [1] Mark Pesce, VRML-Browsing & building cyberspace, New Riders. 1995.
- [2] The moving worlds VRML 2.0 specification. 1996.
<http://vrml.sgi.com/moving-worlds/index.html>
- [3] Cosmo Player The Definitive VRML 2.0 Browser
<http://vrml.sgi.com/cosmoplayer/>
- [4] 김영경, 정문렬, 백두원, 김동현. "3차원 공간 탐색을 위한 헬리콥터 조종사 메타포어와 그 구현", 한국컴퓨터 그래픽스 학회지, vol. 3. No. 1, pp. 57n~67. 1997.
- [5] Chen, Mountford, Sellen. (1988). "A Study in Interactive 3D rotation Using

2D Control Devices”, Computer Graphics, 22(4), 121~128.

[6] Neilson, and Olsen. (1986) “Direct Manipulation Techniques for 3D Objects Using 2D Locator Devices”, Proceedings of the 1986 workshop on Interactive 3D Graphics, 175~82.

[7] Ware and Jessome. (1988). “Using the Bat : A Six Dimensional Mouse for Object Placement”, IEEE CG & A, Nov. 1988, 65~70.

[8] Slater and Davison. “Liberation from Flatland : 3D Interaction Based on the Desktop Bat”, Eurographics '91, pp. 209~221.

[9] Gleicher and Witkin. “Through-the-Lens Camera Control”, Computer Graphics 26(2). pp. 331~340. July 1992.

[10] Mackinlay, Card, and Robertson.

“Rapid Controlled Movement Through a Virtual 3D Workspace”, Computer Graphics. Vol 24, No. 4, 1990.

[11] He, Cohen, and Salesin. “The Virtual Cinematographer : A Paradigm for Automatic Real-Time Camera Control and Directing”, Computer Graphics Proceedings, 1996. pp. 217~222.

정 문 렬



학대학 컴퓨터학부 조교수

- 1980 서울대학교, 계산통계학 학사
- 1982 한국과학기술원 전산학 석사
- 1982~1986 국방관리연구소 연구원
- 1992 Univ. of Pennsylvania 전산학 박사
- 1992~1994 일본 구주공업대학교 지능정보공학과 조교수
- 1994~현재 숭실대학교 정보과

● 제15회 정보산업리뷰심포지움 ●

- 일 자 : 1997년 12월 9일(화) 13:30
- 장 소 : 한국과학기술회관 대회의실
- 주 제 : ‘전자상거래시대의 정보산업 대응전략’
- 주 최 : 한국정보과학회
- 문 의 처 : 한국정보과학회 사무국
Tel. 02-588-9246