

메모리 계층 구조를 사용한 타일 기반 레스터라이저 설계 A Design of a Tile Based Rasterizer Using Memory Hierarchy Structure

김도현*, 곽재창*

Do Hyun Kim*, Jae Chang Kwak*

Abstract

This paper proposes a design of efficient hierarchy structure in the tile based rasterizer. The proposed hierarchy structure avoids unnecessary calls of low level tile at which a calculation is not required. A low level tile is classified into three categories based on its maximum, minimum position, and inside outside test. The necessity of calculations on the corresponding low level tile can be determined by its classification. The overall amount of computations for graphic processing can be reduced by not calling for the low level tile with no calculation. The proposed hierarchy structure can reduce an execution time of graphic processing. It shows higher efficiency with the more vertex density of formulating 3D model.

요약

본 논문은 타일 기반 레스터라이저에서 연산이 필요하지 않은 하위 계층에 대한 호출을 막아 연산의 효율을 올릴 수 있는 계층 구조의 설계를 제안한다. 제안하는 계층 구조는 내외부 판정과 각 하위 계층이 가지는 타일의 최대 좌표값, 최소 좌표값을 이용하여 하위 계층을 3가지 형태로 분류한다. 각 하위 계층이 분류되는 형태에 따라 해당 계층의 연산의 필요 여부를 구분할 수 있으며 연산이 필요하지 않는 하위 계층에 대한 호출을 수행하지 않는 것으로 그래픽 처리과정의 전체 연산량을 줄일 수 있다. 제안하는 구조를 이용하여 하위 계층의 분류를 통해 그래픽 처리의 연산 시간을 줄일 수 있으며 3D 모델을 구성하는 정점의 밀집도가 클수록 높은 효율을 보인다.

Key words : Rasterizer, tile-based rendering, Hierarchy structure, Inside outside test, Coner

* Telechips Inc.

★ Corresponding author

Dept. of Computer Science, Seokyeong University

e-mail : jckwak@skuniv.ac.kr

Tel : 02-940-7758

Manuscript received Dec 15, 2015; revised Dec 28, 2015 ;
accepted Dec 28, 2015

This is an Open-Access article distributed under the
terms of the Creative Commons Attribution
Non-Commercial

License(<http://creativecommons.org/licenses/by-nc/3.0>
) which permits unrestricted non-commercial use,
distribution, and reproduction in any medium,
provided the original work is properly cited.

1. 서론

3D 그래픽은 최근 더욱 화려한 효과와 사실적인 출력 영상을 위해 더 많은 연산량을 요구하게 되었고 이를 처리하기 위해 높은 연산 속도와 효율적인 처리 구조를 가지는 다양한 3D 그래픽 처리 장치가 연구되고 있다. 3D 그래픽의 수요가 점점 늘어나고 특히 휴대 가능한 모바일 기기에서의 3D 그래픽에 대한 요구가 급증하고 있어 제한된 자원에서 동작할 수 있는 구조의 처리 장치와 3D 그래픽의 연산량을 줄일 수 있는 알고

리즘의 연구 또한 큰 이슈로 대두되고 있다.

Rasterization은 3D 그래픽 파이프라인 중 화면을 구성하는 화소와 화소의 색상을 생성하는 과정이다. Rasterization의 연산 과정에서 발생하는 연산의 종속성은 병렬 처리를 활용한 그래픽 처리과정에서 연산 속도를 저하시키는 요인이 되기 때문에 이러한 종속성의 발생을 줄이는 것으로 Rasterization의 성능을 높일 수 있다.

본 논문에서는 병렬 처리를 활용하는 타일 기반 레스터라이저에서 Rasterization의 성능을 높이기 위해 전체 연산량을 줄일 수 있는 계층 구조에 관하여 연구하였다. 본 논문에서는 상위 계층에서 하위 계층을 호출할 때 각 하위 계층의 연산의 필요여부를 판단하여 연산이 필요하지 않은 하위 계층에 대한 호출을 막아 전체 연산량을 줄일 수 있다.

II. 본론

1. Primitive Assembly

화면에 출력되는 화소 정보를 구하기 위해 수행되는 여러 과정은 모두 폴리곤 단위로써 연산을 진행한다. 때문에 Vertex Shading의 결과로 출력된 정점 단위의 정보를 모아 폴리곤으로 구성하는 Primitive assembly 과정을 수행해야 한다[1][2].

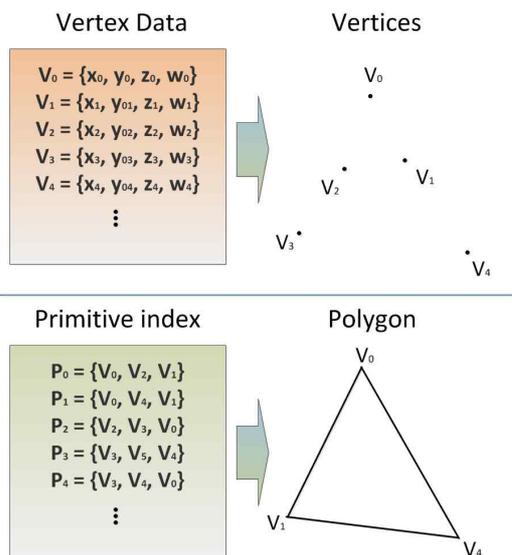


Fig. 1. Vertex Data and Polygon Data
 그림 1. 정점 정보와 폴리곤 정보

폴리곤(Polygon)을 구성하는 최소의 단위는 정점 정보 3개를 이용해 만드는 삼각형(Triangle)으로 이를 기준으로 모든 Rasterizer의 연산을 수행한다. 추가적으로 삼각형 폴리곤을 변형, 조합하여 점(Point), 선(Line), 삼각형 이외의 다각형에 대한 연산을 수행할 수 있다.

2. 화소의 내외부 판정

하위 계층에 대한 분류를 하기 위해 다음과 같은 화소의 내외부 판정이 필요하다[6].

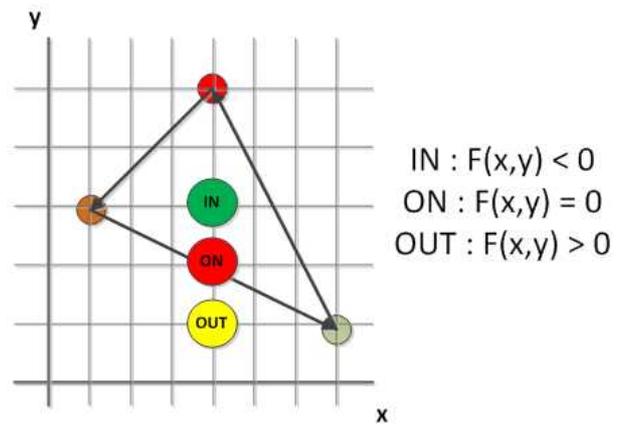


Fig. 2. Inside Outside Test
 그림 2. 내외부 판정

그림 2에서 $F(x,y)$ 는 폴리곤을 구성하는 직선의 방정식($0=ax+by+c$)이며 IN, ON으로 명시된 화소를 폴리곤 내부의 화소로 판단한다.

3. 하위 계층 호출 방법

타일 기반 알고리즘을 통하여 분할된 하위 계층(이하 타일)에 대한 연산을 수행할 때 폴리곤 내부의 화소 외의 모든 화소에 대한 연산을 수행하는 것은 Rasterizer의 성능 저하의 요인이 되기 때문에 타일의 분류를 통하여 연산이 필요하지 않은 타일의 호출을 줄이는 것으로 Rasterizer의 성능 저하를 막는다.

가. 타일의 분류

전체 타일 중 내부에 폴리곤을 포함하여 실제 연산이 필요한 화소가 존재하는 타일과 폴리곤 외부에 있어 연산이 필요하지 않은 타일을 분류하는 것으로 성능 저하를 막을 수 있다.

다음은 본 논문에서 타일을 분류하는 3가지 형

태이다.

- 첫 번째, 타일 전체가 폴리곤 내부에 위치한다.
 - 두 번째, 폴리곤의 일부가 타일 내부에 위치한다.
 - 세 번째, 폴리곤의 어느 부분도 타일 내부에 위치하지 않는다.
- 여기서 타일이 첫 번째와 두 번째에 해당하는 경우 이를 연산이 필요한 타일으로써 분류하여 호출을 수행한다.

나. 분류를 위한 타일의 필요 요소

본 논문에서 타일을 분류하기 위해서는 다음 그림 3과 같은 요소들이 필요하다.

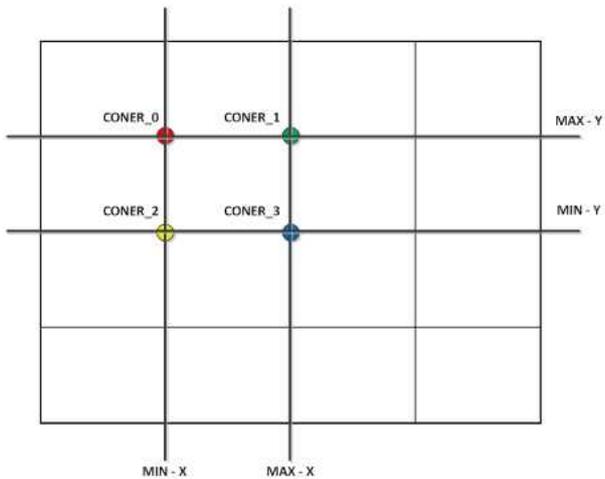


Fig. 3. Tile's Parameter for classification

그림 3. 분류를 위한 타일의 요소

상위 계층으로 사용되는 타일은 화면을 x축으로 4등분, y축으로 4등분하여 하위 계층인 타일을 총 16번 호출할 수 있다. 이 때 상위 계층은 각 하위 계층의 좌상단 좌표, 우상단 좌표, 좌하단 좌표, 우하단 좌표(이하 코너)와 x 좌표의 최대값, 최소값, y좌표의 최대값, 최소값을 이용하여 연산을 수행할 하위 계층과 연산이 필요하지 않은 하위 계층을 구분한다.

다. 연산이 필요한 타일의 분류

타일의 분류에서 첫 번째와 두 번째의 경우 내부의 폴리곤이 존재하여 연산이 필요한 타일으로써 분류된다.

(1) 타일이 폴리곤 내부에 있는 경우

다음 그림 4는 타일이 폴리곤 내부에 존재하는 경우를 나타내는 것이다.

그림 4에서 사선으로 표현된 타일(이하 사선 타일)은 타일 전체가 폴리곤 내부에 있는 첫 번째의 경우로써 4개의 코너를 이용하여 분류할 수 있다.

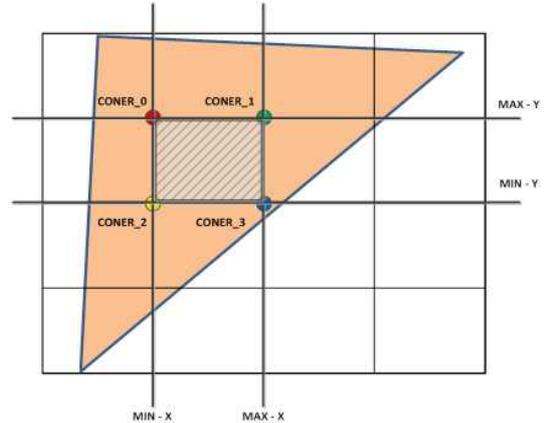


Fig. 4. Tile positioned within the polygon

그림 4. 폴리곤 내부에 위치하는 타일

사선 타일의 모서리에 해당하는 4개의 코너에 대하여 내외부 판정을 수행하고 4개의 코너가 모두 폴리곤 내부에 위치할 경우 사선 타일은 폴리곤 내부에 있다고 판단한다.

(2) 폴리곤의 일부가 타일 내부에 위치하는 경우

다음 그림 5는 폴리곤의 일부가 타일 내부에 위치하는 경우를 나타낸 것이다.

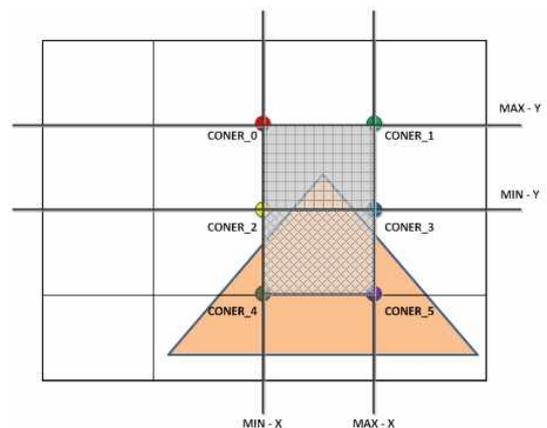


Fig. 5. Tiles comprising a portion of the polygon

그림 5. 폴리곤의 일부를 포함하는 타일

그림 5는 타일 전체가 폴리곤 내부에 포함되지는 않지만 폴리곤의 일부가 타일 내부에 있어 연산이 필요한 타일에 대한 분류이다. 이 때 4개의 코너를 이용한 내외부 판정만으로 ‘타일의 1개 이상 3개 이하의 코너가 폴리곤 내부에 있는 경우 폴리곤의 일부가 타일 내부에 있다’ 라는 조건으로 타일에 대한 검사를 수행할 경우 다음과 같은 오류가 발생할 수 있다.

그림 5에서 CONER_2~5까지로 구성된 사선 격자 무늬의 타일(이하 사선 격자 타일)은 각 코너에 대한 내외부 판정을 수행할 경우 CONER_4, CONER_5가 폴리곤 내부에 포함되어 위의 조건에 부합하게 된다. 그러나 코너 CONER_0~3까지로 구성된 격자 무늬의 타일(이하 격자 타일)의 경우 각 코너에 대한 내외부 판정을 수행했을 때 폴리곤 내부에 포함되는 코너는 하나도 존재하지 않아 위의 조건에 부합하지 않게 된다. 하지만 격자 타일의 경우 분명하게 연산이 필요한 폴리곤의 일부가 포함되어 있기 때문에 위와 같은 조건만으로는 폴리곤의 일부가 포함된 타일을 구분할 수 없다는 것을 확인하였다. 따라서 위의 조건과 함께 타일의 범위를 이용한 수식 1(폴리곤 및 타일의 최대, 최소 좌표 값 비교)의 조건을 함께 사용하여 타일의 분류를 수행한다.

- ① $POLYGON Max X > TILE Min X$
 - ② $POLYGON Min X < TILE Max X$
 - ③ $POLYGON Max Y > TILE Min Y$
 - ④ $POLYGON Min Y < TILE Max Y$
- (1)

수식 1은 모든 코너가 폴리곤 내부에 포함되지 않는 상황에만 적용하면 되므로 조건 검사의 순서는 각 코너의 내외부 판정 이후 수식 1의 조건 순으로 진행한다. 이 순서에 따라 코너의 내외부 판정 결과를 수행한 후 모든 코너가 폴리곤 외부에 위치하는 경우 수식 1의 조건 검사를 모두 수행하고 4가지의 조건을 모두 만족하는 타일은 연산이 필요한 타일로써 분류된다.

그림 5에서 사선 격자 타일은 폴리곤 내부에 CONER_4, CONER_5 두 개의 코너가 포함되어 이미 폴리곤의 일부가 타일에 포함된 연산이 필요한 타일로써 분류가 된다. 그리고 격자 타일은 타일 내부에 폴리곤이 포함되지만 폴리곤 내부에 코너가 하나도 존재하지 않아 수식 1의 조건 검

사를 수행한다. 그림 5의 각 타일의 크기를 1이라 했을 때 수식 1을 다음과 같이 정리할 수 있다.

- ① $POLYGON Max X = 3.x > TILE Min X = 2$
 - ② $POLYGON Min X = 1.x < TILE Max X = 3$
 - ③ $POLYGON Max Y = 2.x > TILE Min Y = 2$
 - ④ $POLYGON Min Y = 0.x < TILE Max Y = 3$
- (2)

수식 2에서 알 수 있듯이 격자타일은 모든 조건에 대해 만족하여 연산이 필요한 타일로써 분류된다.

라. 연산이 필요하지 않은 타일의 분류

타일의 분류에서 세 번째의 경우 내부의 폴리곤이 존재하지 않아 연산이 필요하지 않은 타일로써 분류한다.

다음 그림 6은 타일 내부에 폴리곤이 존재하지 않는 경우를 나타낸 것이다.

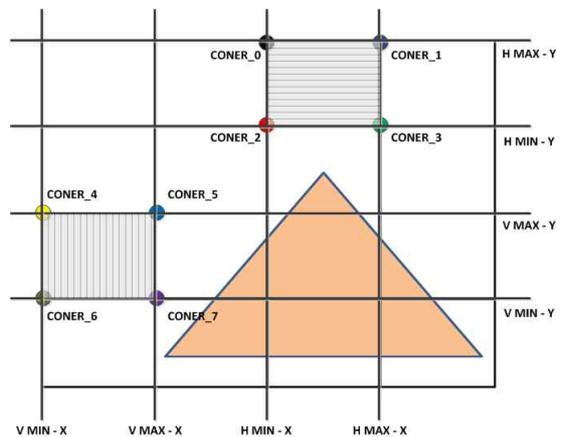


Fig. 6. Tile which is located outside the polygon
그림 6. 폴리곤 외부에 위치하는 타일

타일 내부에 폴리곤이 포함되지 않아 연산이 필요하지 않은 타일의 검사 또한 내외부 판정과 수식 1의 조건을 함께 사용하며 이 때 수식 1의 조건 중 하나의 조건이라도 만족하지 않는다면 해당 타일은 연산이 필요하지 않은 타일로 분류한다.

그림 6에서 각각 가로 선으로 표현된 타일(이하 가로 타일)과 세로 선으로 표현된 타일(이하 세로 타일)은 내부에 연산이 필요한 화소가 존재하지 않는 타일들이다. 이 타일들은 그림 5의 격자 타일과 마찬가지로 각 코너의 내외부 검사에서 모든 코너가 폴리곤 외부에 위치하여 수식 1

의 조건을 검사한다. 격자 타일에서의 조건 검사와 동일한 환경에서 가로 타일과 세로 타일의 수식 1의 조건을 다음과 같이 정리할 수 있다.

$$\begin{aligned}
 & \textcircled{1} POLYGON Max X=3.x > TILE Min X=2 \\
 & \textcircled{2} POLYGON Min X=1.x < TILE Max X=3 \\
 & \textcircled{3} POLYGON Max Y=2.x > TILE Min Y=3 \\
 & \textcircled{4} POLYGON Min Y=0.x < TILE Max Y=4
 \end{aligned}
 \tag{3}$$

$$\begin{aligned}
 & \textcircled{1} POLYGON Max X=3.x > TILE Min X=0 \\
 & \textcircled{2} POLYGON Min X=1.x < TILE Max X=1 \\
 & \textcircled{3} POLYGON Max Y=2.x > TILE Min Y=1 \\
 & \textcircled{4} POLYGON Min Y=0.x < TILE Max Y=2
 \end{aligned}
 \tag{4}$$

수식 3에서 가로 타일은 세 번째 조건을, 수식 4에서 세로 타일은 두 번째 조건을 만족하지 않아 두 타일은 모두 내부에 연산할 화소가 없는 것으로 판단한다.

이러한 분류를 통해 상위 계층은 하위 계층을 호출할 때 연산이 필요하지 않은 하위 계층을 배제하여 불필요한 연산을 줄이는 것으로 성능을 높일 수 있다. 또한 이와 같은 구조를 이용할 경우 전체 화면에서 3D 모델을 구성하는 정점이 높은 밀집도를 지닌다면 연산이 필요한 하위 계층이 줄어들어 더욱 높은 성능을 얻을 수 있다.

4. 실험 및 결과

성능의 비교는 3D 그래픽스의 화소 처리 성능 측정에 가장 널리 사용되는 초당 화소 처리 성능(Pixel Fillrate per second)을 측정하고 이를 비교하였다[4]. 실험을 위해 Virtex-7 XC7VX485T FPGA가 탑재된 Xilinx VC-707 FPGA Board[8]를 사용하여 동작을 검증하였다.

성능 비교를 위해 본 논문에서의 Rasterization 과정과 KAIST의 ‘RAMP’[5]와 ‘병렬처리 보간 기법을 적용한 모바일 GPU용 Rasterizer’[3]를 FPGA에서 동작 시켜 각각의 처리 시간을 비교하였다. 그래픽 파이프라인의 동일한 과정을 진행하기 위해 본 논문에서는 Vertex Shading 과정의 처리 시간을 제외하였고 비교 대상으로 사용되는 두 Rasterizer 또한 Rasterization을 제외한 기능은 동작시키지 않은 결과를 사용하였다.

본 논문 처리 구조가 적용된 Rasterizer의 동작 주파수는 50MHz로 KAIST의 ‘RAMP’와 ‘병렬처리 보간 기법을 적용한 모바일 GPU용 Rasterizer’ 중 KAIST의 ‘RAMP’가 가장 낮은

동작 주파수를 가져 이를 기준으로 세 가지 Rasterizer를 모두 같은 10MHz 환경에서 동작시켰을 때 QVGA 해상도에서 초당 화소 처리 성능을 비교하였다.

Table 1. Rasterizer FPGA Resource

표 1. Rasterizer FPGA 자원 사용량

	KAIST RAMP	Moblie GPU Rasterizer	Proposed Rasterizer
Frequency	10MHz	100MHz	50MHz
Slice Registers	3,431EA	408EA	3,789EA
Slice LUTs	7,471EA	453EA	4,127EA

표 1은 본 논문에서 설계한 Rasterizer와 비교군으로 사용되는 Rasterizer의 FPGA 자원 사용량을 비교한 결과로 Slice Register는 본 논문의 Rasterizer, Slice LUT는 KAIST의 RAMP가 가장 큰 크기를 차지하였다.

Table 2. Comparison - Pixel Fillrate per second

표 2. 초당 화소 처리 성능 비교

Rasterizer	Pixel Fillrate per second
Proposed Rasterizer	63MPixel/sec
KAIST ‘RAMP’	21MPixel/sec
Mobile GPU Rasterizer	53MPixel/sec

표 2에서 KAIST의 ‘RAMP’는 21MPixel/sec, 모바일 GPU용 Rasterizer는 53 MPixel/sec를 보였다. 그리고 본 논문에서의 처리속도는 63 MPixel/sec로 비교 대상의 두 Rasterizer보다 각각 3배, 1.18배 빠른 처리성능을 보이는 것을 확인하였다.

III 결론

본 논문에서는 병렬 처리를 활용하는 타일 기반 레스터라이저에서 Rasterization의 성능을 높이기 위해 전체 연산량을 줄일 수 있는 계층 구조에 관하여 연구하였다.

연구한 계층 구조는 상위 계층에서 하위 계층을 호출할 때 각 하위 계층의 연산의 필요 여부를 판단하여 연산이 필요하지 않은 하위 계층에

대한 호출을 막는 것으로 그래픽 처리 전체의 연산량을 감소시켰다. 이 과정을 통해 Rasterization 연산에서 발생하는 연산의 종속성 줄였고 그래픽 처리 과정의 성능을 높일 수 있었다.

실험 결과 본 논문의 처리 구조를 사용하는 Rasterizer는 선행 연구와 비교하여 1.18배에서 3배까지 처리 성능을 가지는 것으로 나타났다.

타일 기반 레스터라이저에 대하여 본 논문 외에도 다양한 연구가 진행되고 있기 때문에 이후 다양한 관점에서 연구되고 있는 처리 구조들이 상호 연구되어 더욱 높은 성능을 보일 수 있는 구조에 대한 연구가 진행되길 기대한다.

References

- [1] Dong-young Yeo, "A Design of a 3D Graphics pipeline based on Multi-core Processor", The Graduate School of Seokyeong University, 2011.2
- [2] Dool-Bong Jeon, "A Design of Rasterizer including clipping and culling function for a Mobile Graphics", The Graduate School of Seokyeong University, 2008.2
- [3] Jang-seo Ku, "Design of a Rasterizer based on Parallel Processing Interpolation Algorithm for a Mobile GPU", The Graduate School of Seokyeong University, 2013.2
- [4] Jeong-Ho Woo, "Mobile 3D Graphics SoC From Algorithm to Chip", WILEY, 2010
- [5] Ramchan Woo, "Design and Implementation of Low-Power 3D Graphics SoC for Mobile Multimedia Applications", KAIST, 2004.6.
- [6] W.F.P.W. Burgers "Tile-Based Rendering", Master's thesis. Technische Universiteit Eindhoven, Eindhoven, 2005.1
- [7] Woo-Young Kim, "A Design of a Shader based on the Variable-Length Instruction for a Mobile GP-GPU", The Graduate School of Seokyeong University, 2010.2
- [8] Xilinx, "VC707 User Guide", <http://www.xilinx.com>

BIOGRAPHY

Do Hyun Kim (Member)



2014 : BS degree in Computer Engineering, Seokyeong University.
2016 : MS degree in Electronics and Computer Engineering, Seokyeong University.
2015 ~ present : Telechips Inc.

Jae Chang Kwak (Member)



Feb. 1983 BA in Yonsei Univ.
Dec. 1989 MS in Univ. of Iowa
Aug. 1993 PhD in Univ. of Iowa
Mar. 1995 - present Professor at Dept. of Computer Science, Seokyeong Univ.

<Research Interest> Network Traffic Control, QoS, Realtime Scheduling, Embedded System