

디지털 영상의 표현 층위 고도화와 기술 환경

– 90년대 초 고(高) 부동소수점 연산 성능의 보급과
CG/CGI 작업물의 관계를 중심으로

윤나라*

【 차례 】

- I. 여는 말
- II. 하드웨어 성능과 시각 데이터 처리 작업의 관계 : 고(高) 부동소수점 연산 성능 범용 하드웨어의 등장과 전용 하드웨어의 쇠락을 중심으로
- III. 부동소수점 연산 성능의 향상이 디지털 영상 작업 분야에 가져온 변화 : VFX CG/CGI 작업 과정을 중심으로
- IV. 〈터미네이터 2〉, 〈쥬라기 공원〉, 〈공룡 대탐험〉 사례 분석: 디지털 영상의 표현 층위 고도화를 중심으로
 - 1. 〈터미네이터 2(1991)〉 속 액체 금속 로봇의 매끈한 표면
 - 2. 〈쥬라기 공원(1993)〉의 공룡 표현
 - 3. 〈공룡 대탐험(1999)〉의 더 정교하고 세밀해진 공룡 표현
- V. 맺는 말

국문초록

이 글은 융합적 담론의 저변 확대를 통해 인문학 연구자뿐만 아니라 콘텐츠 창작자와 기술 개발 당사자 모두가 공유할 수 있는 공통 기반의 수립에 기여함을 목표로 한다. 이를 위해, 이 글은 창작과 비평, 기술과 예술 모두에서 공히 빈번하게 다루어지는 컴퓨터 그래픽스(Computer graphics, CG) 및 컴퓨터 생성 이미지(Computer generated imagery, CGI) 분야와 이 분야에서 가장 중요한 지표 중 하나인 부동소수점 연산 성능

* 단독저자, 프랑스 파리 8대학 예술공학(ESTA) 박사, 전남대학교 아트&디자인테크놀로지 협동과정 강사, narayoon@chonnam.ac.kr

을 중심으로 융합적 논의의 가능성을 타진한다. 디지털 영상 작업은 가장 바깥쪽의 시각적 층위 한 겹을 제외하면 표현식의 작성과 조작, 즉 수학적 데이터로 다루는 행위이며, 따라서 데이터의 정확도 및 정밀도와 밀접한 관련이 있다. 소수점이 유동(流動)하는 부동소수점 방식은 고정소수점 방식에 비해 더 적은 자릿수만으로도 더 넓은 범위의 수를 더 정확하게 더 정밀하게 표현할 수 있지만, 소수점의 유동성으로 인해 더 높은 연산 성능을 필요로 한다. 1990년대 초반 저비용 고성능의 범용 하드웨어의 등장과 보급은 새로운 기술 환경을 조성했고, 이는 곧 디지털 영상 작업물의 고도화뿐만 아니라 대중화도 함께 견인했다. 이 글은 당시의 상황을 기술과 예술의 융합적 관점을 통해 재조명하고자 한다. 이 과정에서, 이 글은 공학과 수학의 전문 용어를 사용하는 대신 ‘글(또는 말)로 풀어 설명하기’의 방식을 채택할 것이며, 그렇게 함으로써 인문학의 방식으로 문화예술과 정보기술의 상호작용에 관한 실질적인 담론이 가능한지를 실험한다.

열쇠어 : 정보기술, 부동소수점, 디지털 영상 작업, 컴퓨터 그래픽스, 컴퓨터 생성 이미지, 학제 간 융합 연구

I. 여는 말

의사소통 실패는 단순 언어적 문제가 아니며 문제를 일으키는 용어의 정의를 다시 규정하는 정도로 해결할 수도 없다. [...] 의사소통에 실패한 당사자들이 할 수 있는 일은 번역가가 되는 것이다. [...] 각자는 상대방의 이론과 그것의 경험적 귀결들을 자신의 언어로 번역하는 법을 배움과 동시에 그 이론이 적용되는 세계를 자기의 언어로 서술하는 법도 함께 배운다.¹⁾

토머스 쿤(Thomas S. Kuhn, 1922-1996)은 그의 저서 『과학 혁명의 구조(The Structure of Scientific Revolution, 1962)』에서 통약불가능성(通約不可能性, Incommensurability)과 통약불가능한 개념들을 ‘번역’하는 작업의 중요성을 강조한 바 있다. 학제간 연구(學際間研究, Interdisciplinary study)는 통상적 관점에서 상호 이질적인 것 - 혹은 쿤의 용어를 빌리자

1) Thomas S. Kuhn, *The Structure of Scientific Revolutions*, Chicago (United States): The University of Chicago, 1962; 1970 (for 2nd edition), pp.201~202.

면 상호 통약불가능한 것 - 으로 간주되는 둘 또는 그 이상의 분야들을 융합적인 관점에서 접근하는 연구 경향을 아울러 지칭한다. 이러한 맥락을 고려할 때, 학제간 연구자에게 요구되는 가장 기본적이며 필수적인 역량 중 하나는 한 분야의 고유한 성취를 그 분야에서만 통용되는 개념이나 용어를 최대한 적게 사용하면서 다른 분야로 소개하고 도입함으로써 해당 분야들을 한데 아우르는 역량, 즉 ‘번역’ 역량이다.

번역이 언어적(Linguistic) 관점에서 매우 중요한 개념임에 이견을 갖기는 어렵다. 그러나 예전이라면 사진이나 영화, 최근이라면 인터넷이나 메타버스(Metaverse)처럼 기존의 판도를 완전히 뒤집을 수 있을 만큼 획기적이고 새로운 문화기술들의 등장 초기에 우리에게 가장 필요한 직접적이고 정확한 이해는 단지 언어적 측면만으로는 구성 불가능하며, 실제적이고 실질적인 이해가 함께 동반되어야만 비로소 구성 가능하다.

그럼에도 불구하고, 메타버스를 비롯한 동시대적 기술과 기술 환경에 관한 논의의 대부분은 언어적 측면에 편중되어 있다. 가령 어떤 기술을 일컫는 명칭이나 용어의 어원은 무엇이며, 누가 언제 어떤 방식으로 처음 만들었는지, 누가 언제 어느 저서에서 어떤 방식으로 처음 사용했다는 점 등을 강조하는 방식, 즉 실상은 그다지 새롭지도 않은 신조어(Neologism)를 중심으로 하는 방식이 그것이다.²⁾

언어에 기댄 논의는 다양한 해석과 견해의 가능성을 제시한다는 점에서 분명 나름의 가치를 갖는다. 그러나 사진(Photography)이라는 용어가 빛(光, Photo-)을 포함하고 있음을 강조한다고 해서 저절로 사진이라는 매체와 그것의 핵심 구성 요소인 광학(光學, Optics)을 전문적으로 이해하고 다룰 수 있게 되지는 않는다. 마찬가지로, 셀룰로오스 박막(Film)과

2) 영화기호학자 크리스티앙 메츠(Christian Metz, 1931-1993) 역시 신조어에 관한 회의적 입장을 표방한 바 있다. “비평 이론에서 사용되는 각종 신조어들이 진정으로 필수불가결한지에 관해서 개인적으로 확신이 없다. 필수불가결한 것은 이 용어를 통해 드러내고자 하는 개념이기 때문이다.” Christian Metz, *L'Énonciation impersonnelle, ou le site du film*, Paris (France): Méridiens Klincksieck, 1991, p.115. 번역 과정에서 필자가 다소 의역했음을 밝힌다.

영화(Film)의 관계든 움직임(Kine-)과 영화(Cinema)의 관계든 움직임(Movement)과 영화(Movie)의 관계든, 영화에 관련된 용어들에 관한 이해 및 그것들끼리의 관련성을 말과 글로 기술(記述, Describe)하는 것은 매체와 기술 그리고 영화 산업 전반에 관한 이해의 도입부가 될 수는 있으나 본론으로는 적합하지 않다. 같은 맥락에서, 팀 버너스-리(Tim Berners-Lee, 1955-현재)가 누구인지 아는 것이나 “인터넷이란 네트워크들의 네트워크다(The Internet is a network of networks)”³⁾라는 표현 등을 아는 것은 인터넷에 관한 어떤 이야기를 시작하고 이어나가는 데에는 도움이 될 수 있지만 인터넷이라는 기술 환경 자체에 관한 실질적 이해에는 그다지 도움이 되지 않는다.

처음 등장했을 때부터 지금까지 수십 년이라는 꽤 오랜 기간 동안 꾸준히 ‘새로운’ 미디어(New media)로 분류되고 있는 컴퓨터 기반 미디어(computer-based media)들을 둘러싼 논의들 또한 마찬가지다. 이러한 논의들 중 대부분은 해당 기술과 매체에 관한 직접적이고 실질적인 담론들이라기보다 언어적 전개 방식에 기댄 담론에 관한 담론, 즉 메타(Meta)담론인 경우가 빈번하다. 이러한 메타 담론들 중 새로이 등장한 문화 현상들을 그것이 기반을 둔 새로운 기술 환경 및 이를 견인한 핵심 기술들과 함께 충분한 심도로 다루는 융합적 담론을 찾는 일은 쉽지 않다. 반대로 마찬가지다. 어떤 기술이나 기술 환경을 주요 쟁점으로 삼는 담론은 많다. 그러나 그것과 그로 인해 대두된 새로운 문화 현상들을 해당 기술에 관한 관점과 인문학의 관점을 한데 아우른 융합적 관점으로 균형 있게 다루는 담론은 드물다.

최소한 유사 이래로 - 어쩌면 선사 때부터 - 기술과 기술 환경 그리고 문화예술 분야는 긴밀하게 상호작용해 왔다.⁴⁾ 이 상호작용을 논하기 위

3) Robert T. Braden (Ed.), *Requirements for Internet Hosts - Communication Layers*, <https://doi.org/10.17487/RFC1122>, RFC Editor, p.6.

4) 이에 관한 심도 있는 사유는 다음 책을 참고하기 바란다. Friedrich Kittler, *Optische Medien: Berliner Vorlesung 1999*, Berlin (Germany): Merve, 2002. 『광학적 미디어:

해 선사 때의 동굴 벽화와 그을음, 침에 갠 황토, 그리고 손가락의 관계까지 거슬러 올라갈 필요는 적다. 기술과 문화예술의 긴밀한 관계는 논의의 범위를 필름이라는 아날로그 매체 기반 영화와 파일이라는 디지털 매체 기반 영화 둘만으로 한정할 사례 하나만으로도 분명하게 드러난다. 가령 피사체(被寫體)만 놓고 보더라도 그렇다. 아날로그 영화는 어떤 식으로든 물질적(物質的, Material) 또는 물리적(物理的, Physical) 피사체가 필수불가결한 반면 디지털 영화는 그렇지 않다. 1990년대 초반 정보기술 하드웨어 및 소프트웨어의 발전은 컴퓨터 그래픽스(Computer graphics, CG) 및 컴퓨터 생성 이미지(Computer-generated imagery, CGI) 기술의 출현과 성장을 견인했고, 꾸준한 성장 끝에 디지털 영화는 사실적(寫實的, Photorealistic) 혹은 파생실재적(派生實在的, Hyperreal) 가상(Virtual) 피사체를 자유자재로 활용할 수 있게 되었다. 전적으로 가상 피사체만으로 구성된 Full-CG/CGI 영화가 등장한지도 벌써 30년이 넘었으며, 그 이후로 지금까지 여러 Full-CG/CGI 영화들이 대중적·상업적으로 큰 성공을 거두었음은 익히 알려진 사실이다.

통상, 디지털 영상의 표현 층위에 관한 연구는 그것을 표현하는 과정과 원리보다 이미 표현된 것에 중점을 둔 시각 기호 분석과, 내러티브를 중심으로 한 내용 분석이 주를 이룬다. 그러나 디지털 영상 작업은 가장 바깥쪽의 시각적 표현 층위 단 한 겹을 제외하면 표현식의 작성과 조작, 즉 수학적 데이터를 다루는 행위다. 따라서 최외곽 피상 층위뿐만 아니라 그것을 가능케 하는 데이터의 정확도 및 정밀도를 함께 고려할 필요가 있다. 그 첫 걸음으로서, 이 글은 CG/CGI 분야의 변화와 발전에 기여한 여러 요인들 중 하나인 1990년대 초반 고성능 하드웨어의 대중화를, 그 중에서도 이것이 견인한 고(高) 부동소수점 연산 성능의 보급을 기술·창작의 융합적 관점에서 재조명하고자 한다. 이 글은 부동소수점 개념을 정보기술이 아닌 다른 분야의 방식과 언어로 옮겨 소개하고, 해당 개념

1999년 베를린 강의』, 윤원화 역, 현실문화, 2011.

의 중요성을 예증하는 대표적 CG/CGI 작업들을 사례로 살펴봄으로써
기술과 창작 사이의 지속적이고 긴밀한 상호작용에 집중한다.

Ⅱ. 하드웨어 성능과 시각 데이터 처리 작업의 관계 : 고(高) 부동소수점 연산 성능 범용 하드웨어의 등장과 전용 하드웨어의 쇠락을 중심으로

1981년 11월에 문을 연 실리콘 그래픽스(Silicon Graphics, Inc., SGI)는 <터미네이터 2(Terminator 2: Judgment Day, 1991)>나 <쥬라기 공원(Jurassic Park, 1993)> 등으로 대변되는 90년대 초반 영화와 디지털 시각 효과(Visual effects, VFX), 더 정확히 말하자면 VFX CG/CGI를 다루는 담론에서 종종 등장했던 회사였다. 당시 CG/CGI 작업에 SGI제 하드웨어와 여기서 구동되는 소프트웨어는 사실상 업계 표준이나 마찬가지였을 정도로 필수였다. 이에 힘입은 SGI는 80년대 후반과 90년대 초반에 전성기를 구가했지만, 얼마 지나지 않아 사세가 크게 기울기 시작했다. 1995년에 주당 45달러를 상회하던 SGI의 주가는 단 10년 후인 2005년에 주당 46센트가 되었고, 이미 회생 불가능 수준이었던 SGI는 오래 버티지 못하고 2009년 5월 공식적으로 파산했다.⁵⁾ SGI의 쇠락은 이 회사가 컴퓨터 그래픽스 및 컴퓨터 생성 이미지 분야에 기여한 공로만큼이나 중요한데, 여기에는 여느 다른 회사들처럼 경영진의 시대착오적 의사결정이나 ‘혁신가의 딜레마’ 따위가 아니라 완전히 새로운 기술과 기술환경의 출현이 더 결정적인 요인으로 작용했기 때문이다.⁶⁾

이를 이해하기 위해서는 SGI와는 정반대의 위치에서 마찬가지로 중요

5) Randall Hand, “What Led to the Fall of SGI”, *VizWorld*, 2009년 4월-5월 (6부작 심층 분석 기고문). <https://vizworld.com/tag/sgi-bts> (검색 일자 : 2022년 3월 12일).

6) 이에 관한 심도 있는 논의는 앞의 각주에서 제시한 6부작 심층 분석 기고문을 참고하기 바란다. *Ibid.*

한 정보기술 관련 회사인 인텔(Intel corporation)을 살펴볼 필요가 있다. 인텔은 1993년 3월에 발표한 중앙처리장치(Central processing unit, CPU)에 그 때까지 써오던 286, 386, 486 등 종전까지의 이른바 ‘x86’ 명명법을 답습한 586 대신 펜티엄(Pentium™)이라는 이름을 붙인다. 펜티엄이 정보기술의 역사에 남긴 족적과 성과는 이 CPU의 새로운 이름이 아니라 이것이 일반 소매용 대량 생산 제품이었다는 점에 있다. 펜티엄과 관련한 정보기술적 성과는 단지 명칭만을 나열하는 데에도 여러 페이지가 필요할 정도인 관계로, 이 글에서는 그 중 VFX CG/CGI 작업에 가장 관련한 부동소수점(浮動小數点, Floating-point, FP) 연산 성능의 향상만을 중심으로 논지를 전개하기로 한다. 왜냐하면 이것이야말로 SGI로 대표되는 고가의 전용 시스템 및 시스템 제조사들의 쇠락을 견인한, 그리고 VFX CG/CGI 분야를 오늘날의 모습으로 자리 잡게 한 결정적 요인 중 하나이기 때문이다. 특히 병렬 컴퓨팅(Parallel computing), 빅 데이터(Big data) 수집, 빅 데이터베이스(Big database) 구축 및 운용, 그리고 동시대 인공지능(Contemporary AI)의 훈련 및 학습 등 우리 시대 가장 중요한 기술들이 높은 부동소수점 연산 성능에 크게 기대고 있다는 점을 고려하면, 더 늦기 전에 이 개념을 한 번 짚은 짚고 넘어갈 만하다.⁷⁾

부동소수점 개념을 조금이라도 더 실질적인 차원에서 이해하기 위해서는 정보기술 영역에서 사용하는 수 표현⁸⁾ 방식에 대한 이해가 선행되어야 한다. 거칠게 요약하자면, 정보기술 영역에서 사용하는 수 표현 방식은 정수(整數, Integer)부와 소수(小數, Decimal)부를 구분하는 소수점

7) 이 글에서 다룬 범위 외에 부동소수점에 관한 추가 이해가 필요한 경우 다음을 참조하기 바란다. Institute of Electrical and Electronics Engineers(IEEE), “IEEE Standard for Floating-Point Arithmetic” in *IEEE Standard 754-2019 (Revision of IEEE 754-2008)*, <https://doi.org/10.1109/IEEESTD.2019.8766229>, July 2019, pp.1~84.

8) 여기서 ‘표현’이란 모니터 화면 등에 숫자를 표시하는 피상적 층위에 관한 것이 아니라 보다 본질적인 층위, 즉 CPU, RAM, GPU 등의 핵심 하드웨어가 실제로 다루는 데이터 층위에 관련된다. 이는 정보기술의 기본 구조에 관한 논의로서, 이 글의 사용 범위를 벗어나므로 배제한다.

(小數点, Decimal point)이 기능하는 양상에 따라 고정소수점(Fixed point) 방식과 유동 또는 부동소수점(Floating point) 방식으로 구분할 수 있다.

먼저 고정(Fixed)소수점 방식은 전체 자릿수 및 정수부와 소수부의 자릿수 규격을 사전에 미리 정의하는 방식으로, 한 번 정해진 자릿수는 처음으로 되돌아가 다시 정의하기 전까지 바꿀 수 없다. 가령 수를 표현하는 데 총 열여섯 자리를 쓸 수 있고, 프로그래머가 이를 정수부에 일곱 자리, 소수점에 한 자리, 그리고 소수부에 여덟 자리로 나누어 할당했다고 가정하자. 간단한 수준의 유리수(有理數, Rational number)라면 이 정도로 충분하다. 그러나 0.333...처럼 무한히 진행되는 순환소수(循環小數, Repeating decimal number)만 되어도 금세 한계에 봉착하고,⁹⁾ 원주율(圓周率, π [Pi]; 3.1415926535897932384626433832...)이나 자연로그의 밑(또는 자연상수, e ; 2.7182818284590452353602874713...)처럼 일상에서 자주 사용하는 무리수(無理數, Irrational number)마저도 소수점 이하 여덟 자리까지 - π 는 3.14159265까지, e 는 2.71828182까지 - 밖에 표현할 수 없으므로 충분한 수준의 정확도와 정밀도를 확보할 수 없게 된다.¹⁰⁾ 소수점 이하 여덟 자리 다음은 잘라내어 정해진 자릿수에 끼워 맞추는 셈이다.

정수부에는 단 한 자리만 주고 나머지 열네 자리를 전부 소수부에 할당한다면 적어도 π 나 e 같은 무리수에 한해서는 상황을 개선할 수 있다. 그러나 이 경우 16,777,216처럼 비교적 큰 자연수(自然數, Natural number)를 전혀 다룰 수 없게 되고, 큰 자연수와 작은 무리수를 동시에 다루어야 하는 상황에도 전혀 대응할 수 없게 된다. 자릿수를 열여섯이 아니라 서른둘이나 예순넷 정도로 처음부터 넉넉하게 확보해 두는 것은

9) 순환소수 표기법을 도입하면 3 위에 점을 찍어 $0.\dot{3}$ 으로 표시할 수 있으나, 이는 편의를 위함이지 정확도를 위함은 아니며 정보기술의 방식에 부합하지도 않으므로 이 글에서는 다루지 않는다.

10) 이 글은 수 체계에 관한 글이 아니며 수학에 관한 글은 더욱 아니므로 논의를 허수(虛數, Imaginary number)나 복소수(複素數, Complex number) 혹은 그 이상으로 확대하지 않는다.

미봉책에 지나지 않는다. 이러한 방식은 데이터의 덩어리를 불필요하게 증대하여 결국 시스템의 효율을 크게 저하시킬 뿐만 아니라 이 방식을 도입한들 큰 수와 작은 수를 동시에 충분한 만큼의 정확도와 정밀도로 다룰 수 있게 된다는 보장도 없기 때문이다. 상황에 따라 유연하게 대처하는 것이 최선일 테지만, 주지하다시피 고정소수점 방식은 한 번 정해진 자릿수를 도중에 바꿀 수 없다.¹¹⁾

부동소수점 방식의 핵심은 바로 이 ‘유연함’이다. 물에 뜰 부(浮)는 영어 단어 Float를 직역한 것으로, 소수점이 마치 물에 ‘떠다니는’ 것처럼 숫자들 사이를 떠다니는, 즉 유동(流動)하는 것처럼 보인다고 해서 붙인 표현을 그대로 옮긴 것이다. 부동소수점 방식은 전역적인(Global) 차원에서 미리 정의된 자릿수는 없는 대신 각각의 수가 고유하고 독립적인 방식으로 개별 자릿수를 갖는다. 부동소수점의 실제 표현식을 있는 그대로 이해하기에는 지면의 한계를 비롯해 많은 어려움이 있으므로 앞에서 고정소수점을 설명한 방식으로 대신한다.

앞의 경우에서와 마찬가지로 수를 표현하는 데 총 열여섯 자리를 쓸 수 있다고 가정하자. 부동소수점 방식에서는 가장 먼저 소수점이 그 중 한 자리를 가져가는데, 이 때 그 소수점이 주어진 열여섯 자리 중 어느 자리에 놓이는지도 함께 결정된다. 결정된 자리에 소수점이 놓이면, 소수점이 놓인 자리에 따라 자동적으로 그 앞은 정수부, 뒤는 소수부가 된다. 이렇게 하면 단 하나의 체계만으로도 π 나 e 처럼 소수부가 길고 값이 작은 무리수는 소수점의 자리를 앞으로 당김으로써 소수부에 할당된 자릿수를 늘려 정확도와 정밀도를 제고하고, 소수부가 짧거나 없고 값이 큰 자연수는 소수점의 자리를 뒤로 밀어서 정수부에 할당된 자릿수를 늘

11) 이는 고정소수점 방식의 단점이자 동시에 장점이다. 연산의 범위가 명확하게 설정되어 있으며 정확도가 상대적으로 덜 중요하다면 굳이 부동소수점 방식을 고수할 필요가 없기 때문이다. 실제로, 고정소수점 방식으로 수를 표현하는 코드는 부동소수점 방식을 이용하는 코드에 비해 상대적으로 더 간결한 구조를 갖기 때문에 원하는 시스템을 더 쉽고 빠르고 간편하게 구축할 수 있으며 작동 속도도 더 빠르다.

림으로써 유연하게 대처할 수 있다. 즉 부동소수점 방식은 자릿수 때문에 전체 체계를 다시 설정하거나 매번 다른 체계를 도입할 필요가 없거나 매우 적다. ‘부동’은 이처럼 소수점이 한 자리에 매여 있지 않고 수마다 다른 자리에 놓이는 모습이 한 걸음 뒤에서 보면 마치 ‘떠다니는’ 듯 보이는 특성을 반영한 용어다. 그리고 이 유연함 덕분에, 부동소수점 방식은 고정소수점 방식에 비해 같거나 적은 자릿수만으로도 상대적으로 넓은 표현 범위와 높은 정밀도를 보장한다.¹²⁾

Ⅲ. 부동소수점 연산 성능의 향상이 디지털 영상 작업 분야에 가져온 변화 : VFX CG/CGI 작업 과정을 중심으로

부동소수점 연산이 386이나 486 등 펜티엄 이전의 CPU에서 아예 불가능했던 것은 아니다. 그러나 그 성능은 ‘가능하다’ 수준에 그쳤을 뿐 실제로 사용하기에는 많은 무리가 있었다. 정식 명칭 대신 통상 ‘수학 보조 프로세서(Math coprocessor)’라는 다소 모호한 이름으로 더 잘 알려진 287, 387, 487 등 고가의 부동소수점 연산 장치(Floating-point unit, FPU)를 추가로 구입하여 장착하면 다소 개선되었으나, 그럼에도 불구하고 실제 업무에 적극적으로 활용할 수 있을 만큼 유의미한 성능이 보장되지는 않았다. 이러한 맥락을 고려하면, 펜티엄이 CPU-FPU 통합 설계 방식으로 제조된 중앙처리장치 중 경쟁력 있는 성능을 제공한 첫 대량 생산 상용 제품이었다는 점은 그 자체로 시사하는 바가 적지 않다. 펜티엄의 부동소수점 연산 성능은 486과 487을 동시에 작동시킨 것보다 항

12) 고정소수점 방식과 정반대의 의미에서, 이는 부동소수점 방식의 장점이지만 동시에 단점이다. 매 데이터마다 소수점의 위치가 상이할 수 있다는 말은 새로 입력받은 데이터에서 소수점의 위치를 찾고 정렬하는 데에 주어진 정보기술 시스템의 연산 성능 일부를 할당하는 것이 불가피함을 의미하고, 이는 전체 가용 연산 성능의 감소로 직결된다. 뿐만 아니라 부동소수점 방식으로 데이터를 다루는 프로그래밍 코드가 고정소수점 방식의 그것에 비해 상대적으로 더 복잡해질 수 있으며, 시스템 구축에 더 많은 노력이 필요할 수 있다는 뜻이기도 하다.

상 빨랐고, 15배 이상 격차가 벌어지는 경우도 빈번했다. 물론 통합 설계 탓에 초기의 펜티엄의 가격은 동일한 작동 주파수(Clock speed)의 486 1개보다는 다소 높게 책정되었지만, 마찬가지로 통합 설계 덕에 486 CPU 1개와 487 FPU 1개를 합한 것보다는 유의미할 만큼 저렴했다.

앞서 언급한 SGI의 제품 포트폴리오는 주로 특정 소수의 전문가나 전문가 집단을 상대로 한 고가의 전용 시스템들로 구성되어 있었다. 펜티엄이 출시되었을 즈음, 상용 등급(Commercial grade) VFX CG/CGI 작업에 주로 사용되는 SGI제 시스템 한 세트의 가격은 최소 10만 달러부터 시작했고 25만 달러를 넘는 경우도 빈번했다.¹³⁾ 이는 번듯한 집 한 채를 담보로 저당 잡혀서 마련해야 할 정도로 많은 비용이었다. 그러나 그만큼 비용을 동원하더라도 겨우 오닉스(Onyx) 한 대와 거기서 돌아가는 플레임(Flame) 앱 한 카피를 간신히 마련할 수 있을 뿐이었고, 설령 어렵사리 한 세트 갖추었다고 하더라도 현업 수준의 미려한 결과물을 생산하는 일은 여전히 요원했다.¹⁴⁾ <쥬라기 공원>이나 <터미네이터 2>같은 상용 등급의 VFX CG/CGI 작업은 여러 세트의 오닉스를 한 데 묶어 동시에 병렬로 작동(Clustering)시켜야 할 정도로 높은 성능을 요구했기 때문이다. 이는 필요비용의 급수적 상승을 의미했고, 그 정도의 비용은 알만한 규모의 제작사들에게도 무시하기 어려운 부담 요인이었지만 마땅한 대체재는 드물었다.

반면 펜티엄의 목표 시장(Target market)은 SGI나 오닉스와 전혀 달랐다. 펜티엄은 불특정 다수의 일반 소비자를 대상으로 한 대량 생산 범용 CPU였고, 이는 펜티엄이 태생적으로 고가일 수 없었음을 뜻한다. 실제

13) Steve Jarratt (Ed.), "Silicon Graphics: showing off" in *Edge* issue 7, Bath (United Kingdom): Future Publishing, April 1994, pp.18~19.

14) 이 시기 CG/CGI작업과 비용의 관계에 관한 추가 정보는 당시 업계 관계자들의 인터뷰를 중심으로 작성된 다음 기사를 참고하기 바란다. Mark Walton, "From Paintbox to PC: How London became the home of Hollywood VFX", *Ars Technica*, 2016년 5월 1일. <https://arstechnica.com/gaming/2016/05/how-london-became-the-home-of-hollywood-vfx> (검색 일자 : 2022년 3월 12일).

로 펜티엄 한 세트의 가격은 아무리 비싸 봤자 오닉스 한 세트 값의 1/10을 넘지 않았지만 성능은 오닉스 여러 대를 묶은 중대형 클러스터 시스템보다 거의 모든 면에서 최소 3배 이상 높았고, 둘의 성능 격차는 전체 작업 흐름(Workflow)에서 부동소수점 연산이 차지하는 비중이 커질수록 더욱 벌어졌다. 간단히 말해서, 펜티엄은 VFX CG/CGI 업계에 고가의 전용 시스템 대비 단지 객관적 절대 성능(Raw performance) 뿐만 아니라 가격 대 성능(Performance per price)마저도 출중한 대체제가 출현했음을 의미했다. 업계는 같은 성능이라면 오닉스 한 세트 대신 펜티엄 한 세트를 구입함으로써 비용을 획기적으로 절감할 수 있거나, 혹은 같은 비용이라면 오닉스 몇 대 규모의 중대형 클러스터 대신 펜티엄 수십 대 규모의 대형 또는 초대형 클러스터를 구축하여 작업 기간을 획기적으로 단축할 수 있는 선택지를 갖게 되었다. 그리고 VFX CG/CGI 업계 또한 다른 모든 업계와 마찬가지로 시간이 곧 비용이며, 비용이 곧 시간이다.

실제 CG/CGI 작업 과정을 들여다보는 것은 높은 정확도와 정밀도가 특징인 부동소수점 개념이 VFX CG/CGI 작업에 미치는 실질적 영향을 이해하는 데 특히 주효하다.¹⁵⁾

- 1) 화면 내에 작업의 기준점을 설정한다.
- 2) 2D CG/CGI 작업의 경우 앞의 기준점을 영점으로 하는 평면 좌표계를, 3D CG/CGI 작업이라면 공간 좌표계를 설정한다. 이 때 설정한 좌표계는 평면기하(平面幾何, Plane geometry), 공간기하(空間幾何, Space geometry), 위상기하(位相幾何, Topology) 등 수학의 하위 분야인 기하학의 기본 법칙들에 의거한다. 좌표계의 규모와 세밀함의 정도는 최종 결과물의 규모와 정밀도에 직결된다.
- 3) 설정된 좌표 평면 상 혹은 좌표 공간 내에 픽셀(Pixel), 복셀(Voxel),¹⁶⁾

15) 참고. John F. Hughes et al., *Computer Graphics: Principles and Practice*, Boston (United States): Addison-Wesley, 2013 (for 3rd edition).

16) 래스터 그래픽스(Raster graphics, 비트맵[Bitmap]과 혼용하는 경향이 점차 증가하고 있으나 둘은 별개의 개념이므로 명확히 구분하여 사용할 필요가 있다) 방식 2D 객체

넵스(Non-uniform rational basis spline, NURBS),¹⁷⁾ 벡터(Vector) 변곡선, 꼭짓점(Vertex), 다각형(Polygon), 질감(Texture) 등을 이용하여 필요한 객체(Object)들을 만들어 넣는다. 용어나 외양의 차원과 무관하게, 본질적 층위에서 모든 객체는 다수의 값(Value), 변수(Variable), 연산자(Operator), 함수(Function)들을 항(項, Term)과 매개변수(媒介變數, Parametre)로 갖는 식(式, Expression)으로 구현된다. 이 때 각 항, 매개변수, 그리고 식의 정밀도는 그것이 만드는 객체의 정교함 정도를 직접적으로 결정한다.

- 4) 광원(光源, Light source)은 빛을 내는 객체로서, 요구되는 경우 광원의 종류, 개수, 위치, 밝기, 이동 경로 등을 정의(Define)하여 앞에서 설정한 좌표 평면 상 혹은 좌표 공간 내에 배치한다. 여타 객체들과 마찬가지로 광원도 본질적 층위에서는 식의 형태로 존재하며, 3에서와 같은 이유로 식의 정교함의 정도는 광원의 크기를 비롯하여 광원이 내는 빛의 직진, 반사, 굴절, 분산, 산란, 간섭, 회절 등 전반적 양태에 직결된다.
- 5) 필요한 경우 3에서 만들어 넣은 객체들을 조작(Operate, pilot, maneuver, manipulate...)할 수 있다. 등장, 이동, 변형, 크기 조절, 퇴장 등 일체의 객체 조작 행위는 곧 해당 객체의 표현식을 구성하는 항과 매개변수를 조절하는 행위에 해당하므로, 이 역시 표현식과 좌표계의 정밀함의 정도와 매우 유관하다.
- 6) 5의 과정은 4에서 만들어 배치한 광원들로부터 나오는 빛과의 상호작용을 필수적으로 수반한다. 가령 이동이나 변형은 그림자를 만들거나 없애고, 표면 질감의 변화는 빛이 반사되거나 굴절되는 방향과 방식

의 단위 구성 요소가 픽셀(Pixel)이듯, 복셀 그래픽스 방식 3D 객체의 단위 구성 요소는 복셀(Voxel)이다. 드물게 이 용어를 복셀(複+Cell)로 오해석하여 픽셀을 겹겹이 쌓은 것(複)이라는 식으로 우회적으로 유비하는 경우가 있는데, Voxel이라는 용어 자체가 Pixel에서 그림(Picture)에 해당하는 ‘Pi-’를 부피(Volume)에 해당하는 ‘Vo-’로 대체하여 만든 것이라는 점을 고려하면 복셀은 겹쳐 쌓아놓은 픽셀이 아니라 부피를 갖는 픽셀로 이해하는 편이 더 타당하다. 용어에 관한 자세한 설명은 앞의 책을 참조하기 바란다. *Op.cit.*, pp.349~350.

- 17) 드물게 ‘비균일 유리 B-스플라인’으로 번역하는 NURBS는 한 개 이상의 자유곡선을 접거나, 구부리거나, 넓게 늘려 퍼거나, 작게 구기는 등 다양한 방식으로 조작하여 필요한 선과 면들을 만든 다음, 마치 조립하듯이 이들을 서로 이어 붙임으로써 최종 결과물을 완성하는 방식이다. 이에 관한 자세한 설명은 앞의 각주에 제시된 문헌을 참조하기 바란다.

등을 바꾼다.¹⁸⁾ 이 또한 궁극적으로는 해당 조명과 객체의 표현식이 상호작용하는 것이므로 정밀도에 크게 영향 받는다.

- 7) 여기까지의 과정을 그대로 최종 결과물로 사용하거나, 적절한 형식으로 저장하여 원하는 방식으로 재가공한 다음 최종 결과물로 다듬어 낸다.



[그림 1] CG/CGI 작업 과정(왼쪽에서 오른쪽으로)에 관한 간략한 사례¹⁹⁾

정리하자면, 정보기술을 활용한 다른 모든 작업들과 마찬가지로 VFX CG/CGI 작업도 가장 바깥쪽의 시각적 층위 한 겹을 제외하면 표현식의 작성과 조작, 즉 수학적 데이터를 다루는 행위인 셈이다.²⁰⁾ 부동소수점 방식은 고정소수점 방식에 비해 더 크고 넓은 범위의 데이터를 더 정확하게, 정교하고, 정밀한 방식으로 표현할 수 있다. 펜티엄으로 대표되는 저가의 일반 소비자 대상 대량 생산 범용 CPU들의 약진은 고속·고정밀 부동소수점 연산 성능에 관한 진입장벽을 대폭 낮춤으로써 기존의 기술 환경에 일대 변화를 불러일으켰다. 접근성이 제고되자 VFX CG/CGI 업계는 기다렸다는 듯이 부동소수점 개념을 현업 전반에 대폭 확대 적용

18) 광원과 객체의 상호작용은 CG/CGI 분야 내에서 가장 활발한 논의가 이루어지는 주제 중 하나다. 그림자 처리(Shading), 광선 추적(Ray tracing)을 비롯한 다수의 개념들이 이와 직접적으로 관련된다. 이에 관한 논의는 후속 연구에서 다루기로 한다.

19) John Lasseter, *Luxo Jr.*, 1986.

20) 이는 레프 마노비치가 제시한 디지털 미디어의 정의와도 연결 가능하다. “컴퓨터에서, 디지털 미디어의 객체(object)는 숫자의 형태(numerical form)로 만들어진다.” 모든 디지털 미디어 객체의 모든 요소(외형, 컬러, 깊이 등)는 기저에서 수학적 함수의 형태로 존재하며, 알고리즘 등을 통해 함수와 함수의 구성 요소들을 조작하는 행위는 곧 객체를 조작하는 것과 같다. 그러므로 디지털 미디어는 기본적으로 “프로그램 가능한 것(programmable)이다.” Lev Manovich, *The Language of New Media*, Cambridge, Massachusetts (United States): The MIT Press, 2001, pp.27~28.

했고, 더 다양한 방식으로 제작된 더 정교한 결과물들을 더 빠른 속도로 더 많이 만들어냈다.²¹⁾

즉 기술과 기술 환경의 관점에서 보면, 90년대 중반 즈음부터 영화 산업이 VFX CG/CGI를 더 적극적으로 활용하게 된 이유는 단지 그것을 적극적으로 활용한 영화들이 상업적으로 큰 성공을 거두는 바람에 영화에 CG나 CGI를 사용하는 것이 이른바 ‘대세’가 되었기 때문이 아니라, 더 정교한 VFX CG/CGI를 전보다 더 쉽게 만들고 활용할 수 있는 기술 환경이 비로소 조성되었기 때문이라고 보는 편이 더 타당하다.²²⁾

IV. 〈터미네이터 2〉, 〈쥬라기 공원〉, 〈공룡 대탐험〉 사례 분석: 디지털 영상의 표현 층위 고도화를 중심으로

90년대 초, 일반적인 상업 영화용 VFX CG/CGI는 TV 방송용의 그것에 비해 아무리 적게 잡아도 평균 열여섯 배 이상 많은 작업량을 요구했다.²³⁾ 전통적으로 산업은 작업의 양을 작업에 드는 시간으로 치환하고, 시간을 다시 인력과 비용 등으로 환산한 다음 적정 수준의 인적·물적 자원(Resource)을 작업에 투입함으로써 문제를 해결한다. 여타 산업 분야들과 마찬가지로 VFX CG/CGI 분야에서도 시간은 가장 중요한 문제였고 대부분의 문제들은 기존 방식으로 해결 가능했지만, 컴퓨터라는 새롭고 낯선 장비는 시간에 관한 새로운 차원의 문제를 제기했다. 작업된

21) 1990년대 초반 정보기술 하드웨어의 성능 향상과 창작의 관계는 매체철학의 관점으로도 사유 가능하다. 일례로, 빌렘 플루서는 창작자 및 작업자, 기술적 장치, 소프트웨어, 사회·경제적 구조 등의 관계에 프로그램(Programm)과 메타프로그램(Metaprogramm)의 개념을 적용한다. 참고. Vilém Flusser, *Für eine Philosophie der Fotografie*, Göttingen (Germany): European Photography, 1983, pp.27~28.

22) 기술 발전에 따른 인식 변화에 관한 심도 있는 논의는 다음을 참고하기 바란다. André Leroi-Gourhan, *Le geste et la parole : la mémoire et les rythmes*, Paris (France): Éditions Albin Michel, 1965; 또는 Bernard Stiegler, *La technique et le temps, Tome 2 : La Désorientation*, Paris (France): Éditions Galilée, 1996.

23) 참고. Mark Walton, *op.cit.*

분량을 최종 형태로 그려내기(Rendering)와²⁴⁾ 그에 드는 시간(Rendering time)이 그것이다.

당시 상용 등급 VFX CG/CGI를 렌더링 하는 일은 ‘시간과의 싸움’ 이
상이었다. 이 글의 앞에서 언급한 SGI의 오닉스 클러스터 등 업계 최고
성능의 시스템을 최적의 상태로 구비하여 동원하더라도, 작업물의 최종
렌더링에는 프레임 당 적게는 몇 시간에서 많게는 며칠이 소요되었다.²⁵⁾
사소한 오류에 렌더링이 강제 중단되어 처음부터 다시 시작하는 경우도
빈번했다. 이는 단 몇 초짜리 짧은 쇼트(Shot) 하나를 만드는 데에 적게
는 며칠에서 많게는 몇 달이 걸린다는 뜻이었고, 그렇게 만들어진 쇼트
에서 수정 사항이 단 하나라도 발생하면 오직 그 수정 사항이 제대로 반
영되었는지 확인하는 데에만 다시 며칠에서 몇 달을 기다려야 한다는 말
이기도 했다. 요컨대 90년대 초중반까지 상업 영화용 VFX CG/CGI 작
업은 투입되는 인적·물적 자원의 양이나 결과물의 완성도와 무관하게
작업의 타당성(Feasibility) 자체를 의문에 부쳐도 이상하지 않을 만큼의
저효율에 시달리고 있었다.

24) 이 과정을 거쳐 최종적으로 완성된 것, 즉 ‘렌더된(Rendered) 것’이 관객, 시청자, 소
비자, 혹은 최종 사용자(End user)가 인지하는 시각 기호다. 그리고 주지하다시피, 전
통적·통상적 관점에서 영상기호학의 범주에 속하는 논의는 주로 이 시각 기호 층위,
즉 최종 결과물 층위에 천착하는 경향이 있다.

25) *Ibid.*

1. 〈터미네이터 2(1991)〉 속 액체 금속 로봇의 매끈한 표면

사실상 VFX CG/CGI산업 자체를 정의했다고 보아도 과언이 아닌 조지 루카스(George Lucas, 1944-현재)의 ILM(Industrial Light & Magic)도 예외는 아니었다. 주어진 작업물을 완성하는 것은 당연한 일이었고, 그것을 빠르고 정확하게 완성하려면 어떻게 해야 하는지, 즉 작업의 효율을 어떻게 제고할 수 있는지가 실질적인 관건이었다. ILM은 SGI 같은 전용 시스템 제조사들의 가장 중요한 고객사 중 하나였기에, 외부적으로는 시스템 제조사들과 긴밀히 협조함과 동시에 내부적으로는 작업에 필요한 소프트웨어들을 직접 개발하는 등 각고의 노력을 기울였지만 한계는 분명했다. 왜냐하면 “소프트웨어는 어떤 행위를 실행에 옮기기 위한 계획과 과정일 뿐, 그것을 실제로 실행하는 것은 하드웨어”²⁶⁾이기 때문이다. 제아무리 잘 작성된 소프트웨어라도 그것을 실제로 실행시킬 하드웨어가 애초에 제공하지도 않는 기능이나 성능을 무(無)로부터 창조해 낼 수는 없었다.²⁷⁾

고성능의 대중화는 VFX CG/CGI 분야가 시달리던 저효율 문제의 해결뿐만 아니라 CG/CGI를 통해 표현 가능한 범위 자체의 확장에도 결정적으로 기여했다. 통상적인 CG/CGI 작업의 경우, 작업 흐름 전반에 걸쳐 가장 큰 변인으로 작용하는 것은 CG/CGI 객체의 복잡도다. 객체의 복잡도는 이 글이 앞에서 언급한 구성 요소들, 즉 픽셀과 복셀, 넘스와 벡터 변곡선, 꼭짓점, 다각형, 질감, 광원, 광원-객체간 상호작용 등의 양과 복잡도에 의해 복합적으로 결정된다. 객체의 복잡도가 증가하면

26) Jack Burnham (Ed.), “Notes on art and information processing”, in *Software* (exhibition catalogue), New York (United States): Jewish Museum, 1970, p.14.

27) 모든 소프트웨어의 목적은 하드웨어를 작동 시키는 것, 키틀러 식으로 말하자면 ‘하드웨어를 구성하는 반도체 소자들에 가해지는 전압을 조작(Manipulate)하는 것’이며, 마찬가지로 모든 하드웨어의 목적은 그러한 소프트웨어를 떠받치는 물질적 기반(Support)을 구성하는 것이다. 하드웨어와 소프트웨어의 상호 의존적·상호 보완적 관계에 관한 키틀러의 관점은 다음을 참조하길 바란다. Friedrich A. Kittler, “There Is No Software” in *Stanford Literature Review* 9:1 (Spring 1992), pp.81~90.

CG/CGI 작업의 난이도가 올라가고, 작업에 필요한 연산량도 함께 증가한다. 이는 충분한 연산량이 확보되어있지 않으면 복잡한 CG/CGI 객체를 만들기 어렵거나, 애초에 작업 개시 자체가 불가할 수도 있음을 의미한다.

이 글의 맥락에서, CG/CGI 작업의 난이도와 연산량은 객체를 구성하는 요소들을 적게 사용할수록, 객체의 수가 적고 크기가 작을수록, 그리고 광원을 비롯한 각 요소 및 객체들의 상호작용의 양과 복잡도가 적을수록 낮아지며 그 반대일수록 높아진다. 가장 단순한 예로, 작업의 난이도와 연산량은 화면 내에 CG/CGI 객체의 수가 적을수록 하락한다. 같은 맥락에서 밝은 낮보다 어두운 밤이 광원과의 상호작용이 적고, 넓고 열린 실외보다 좁고 닫힌 실내가 연산량이 적으므로 구현하기에 더 용이하다. 또한 객체가 한 자리에 고정되어 있으면 이동에 따르는 변화를 연산할 필요가 없기 때문에 더 낮은 연산 성능만으로도 어렵지 않게 구현할 수 있다. 이 때 객체의 외양은 복잡하고 구상적인 것 보다 단순하고 추상적인 것이, 즉 복잡하고 비정형인 자연물보다 단순하고 정형인 인공물이 더 작업하기 쉽고 연산량도 적다. 마찬가지로, 생물의 거칠고 불규칙적인 살갗 표피보다 무생물의 매끈하고 규칙적인 도자기나 플라스틱 또는 금속 표면이 더 쉽게 구현 가능하며, 따라서 사람보다 인형이나 로봇의 CG/CGI 작업이 상대적으로 더 쉬운 편에 속하며 연산 성능도 덜 요구한다.

CG/CGI 객체의 투명도(Transparency)를 결정하는 일 또한 그것의 구현에 요구되는 작업량과 연산량의 관점에서 풀어볼 수 있다. 객체를 투명한 액체로 설정한 경우, 빛이 액체의 경계면을 통과할 때 발생하는 굴절 현상은 물론이거니와 고여 있거나 흐르는 액체 뒤로 비치는 바닥과 뒤편까지 구현해야 하므로 작업량과 연산량이 급증한다. 반면 녹은 초콜릿이나 흐르는 수은처럼 불투명한 액체는 뒤가 비치지 않으므로 가장 바깥쪽 표면을 제외한 나머지 부분의 구현을 최소화할 수 있다. 뒤가 비치는 투명한 유리 창문보다 어느 한 쪽만 비추는 거울의 작업 난이도가 상대적으로 더 낮은 것도 같은 이유다. 이는 연산 성능이 제한된 경우 투

명한 객체보다 불투명한 객체가 더 선호될 수 있음을 의미한다.

CG/CGI 객체의 표면도 같은 맥락에서 이해할 수 있다. 객체의 표면이 ‘가장 바깥쪽의 시각적 층위 한 겹’에 불과하다고 해서 작업의 난이도나 연산량과 관련이 적은 것은 아니기 때문이다. 금속을 예로 들어보자. 오래된 금속의 낡고 녹슨 표면보다 이제 갓 베틀낸 금속의 매끈한 표면에서 빛의 난반사(乱反射, Diffused reflection)가 더 적게 발생한다. 그리고 CG/CGI의 관점에서, 더 적은 난반사는 더 적은 연산량과 더 적은 작업량으로 이어진다.

1991년의 <터미네이터 2>는 ILM의 대표작 중 하나에 속한다. 복도 바닥으로 위장하고 있던 액체 금속이 모양을 갖추는 장면은 이 영화에 사용된 CG/CGI 중 가히 백미(白眉)라고 할 만한 것임과 동시에 연산 성능과 실제 결과물의 관계를 살펴보기에도 적합한 사례다.²⁸⁾



[그림 2] <터미네이터 2>의 액체 금속 CG/CGI²⁹⁾

28) Lev Manovich, *Software Takes Command*, New York: Bloomsbury Academic, 2013, p.58.

29) James Cameron, *Terminator 2: Judgment Day*, 1991.

이 글의 맥락에서, 이 장면에 사용된 CG/CGI는 연산량과 작업량의 절감을 위해 크게 두 개의 상식에 기대다. 하나는 수은(水銀, Hydrargyrum; 원소 기호 Hg)처럼 실제로 상온에서 액체 상태인 금속이 있으며 대개 그러한 액체 금속의 표면은 거울처럼 매끈하다는 것이며, 나머지 하나는 유리처럼 뒤가 비치는 투명(Transparent) 또는 광 투과성(Translucent) 금속은 없다는 것이다. 이 장면에서, 복도 바닥에 깔린 액체 금속 터미네이터는 바닥 무늬를 그대로 투과시키는 방식이 아니라 마치 카멜레온이 주변을 흉내 내듯 복도 바닥의 무늬를 그대로 따라서 복제하는 위장 방식을 취한다. 이 액체 금속의 표면의 매끈한 표면은 그것이 ‘위로 흘러서’ 형체를 만드는 모습을 클로즈업 할 때 그 표면에 비추어지는 반사상(反射像)의 작업량과 연산량을 덜기에 효과적이다. 복도 바닥의 단순하고 반복적인 흑백 무늬도, 단지 일상에서 흔히 찾아볼 수 있어서라기보다 CG/CGI 작업을 용이하게 하기 위한 의도적 선택으로 보는 편이 더 타당하다. 카메라의 움직임과 미장센(Mise-en-scène)도 연산량 절감에 일조한다. 이 장면에서, 카메라는 로봇이 형체를 이루기 위해 바닥으로부터 ‘일어나는’ 것을 보여주는 몇 초짜리 짧은 쇼트를 제외하면 사실상 고정되어 있다고 보아도 무방할 만큼 거의 움직이지 않는다. 이는 상대적으로 느린 리듬을 활용하여 액체 금속 로봇의 위협적 잠재력을 최대한 자세히 드러내 보임과 동시에 CG/CGI의 연산량도 함께 절감하는 섬세하고 경제적인 연출이다. 이렇게 함으로써, ILM은 상대적으로 적은 연산 성능만으로도 <터미네이터 2>의 CG/CGI를 충분히 설득력 있는(Convincing) 수준까지 구현할 수 있었다.

<터미네이터 2>는 여기에 내러티브적 장치를 이용해 CG/CGI에 필요한 연산량을 추가적으로 절감한다. 가령 아무리 거칠게 다루어도 어떤 흠도 낼 수 없을 만큼 단단한 금속으로 만들어진 로봇이 있다가거나, 또는 어떤 흠이라도 아주 짧은 시간 안에 회복하고 복구하는 재질로 된 비인간적 존재가 있다는 식으로 내러티브를 설계하는 것이 이에 해당한다.³⁰⁾

간단히 말해서, 연산 성능이 제한된 상황이라면 사람보다 사람의 걸모습을 한 어떤 다른 존재자가, 생물보다 무생물이, 즉 인간보다 비인간인 CG/CGI로 구현하기에 더 용이하다.

2. <쥬라기 공원(1993)>의 공룡 표현

인간이나 생물 같은 유기체를 CG/CGI로 구현하는 경우, 가장 어려운 도전 지점은 각 유기체의 피부나 가죽, 털 등이 갖는 고유한 질감이다. 이 질감을 충분히 설득력 있는 수준으로 구현하려면 객체의 표면을 무기체인 로봇이나 무생물에 비해 더 거칠게 설정해야 한다. 그런데 거친 표면은 조명을 단순히 반사하는 것에 그치지 않고 굴절·분산·산란·간섭·회절 등 더 다양한 방식으로 조명과 상호작용한다. 그리고 이 상호



[그림 3] <쥬라기 공원>의 공룡 CG/CGI³¹⁾

30) <터미네이터 2>에서 CG/CGI 작업의 주 대상인 액체 안드로이드와 그 상징적 함의에 관해서는 다음 논문을 참고하기 바란다. 이수진, 「현대 사이언스 픽션 영화의 기호학적 읽기 : <터미네이터>의 테크놀로지에 관한 시각적 재현을 중심으로」, 『기호학연구』 44집, 한국기호학회, 2015, pp.203~225.

31) Steven Spielberg, *Jurassic Park*, 1993.

작용의 복잡도는 진화 단계의 관점에서 더 고등 생물에 가까워질수록 증가한다. 즉 깃털을 가진 조류나 털가죽으로 덮인 포유류의 표피는 어류나 양서류, 파충류의 그것에 비해 더 거칠고, 따라서 작업의 난이도가 상승하고 연산량도 증가한다. 시스템의 연산 성능이 상대적으로 낮고 제한되어있던 90년대 초반까지 매끈하고 번들거리는 표면을 한 파충류와 양서류가 VFX CG/CGI의 주류를 이루었던 것은 그러므로 단순히 우연이었거나 대세였거나 유행이었기 때문만은 아니다.

ILM의 또 다른 대표작인 <쥬라기 공원>은 CG/CGI 파충류의 향연이라고 보아도 과언이 아니다. <쥬라기 공원>은 <터미네이터 2>에 비해 더 많은 수의 CG/CGI 객체가 등장하고 객체와 조명의 복잡도도 훨씬 증가했지만, 그럼에도 불구하고 영화 곳곳에서 연산량 절감을 위한 우회책을 확인할 수 있다. CG/CGI의 사용량 자체를 최소화하는 것은 가장 직접적인 연산량 절감책이었다. 이 영화가 당대의 CG/CGI에 관한 담론들에 이름을 올리는 횡수와는 별개로, <쥬라기 공원>에서 CG/CGI가 차지하는 분량은 약 128분의 전체 상영 시간 중 단 6분 남짓으로 채 4.7퍼센트에도 미치지 않는다. 이 영화에 사용된 시각 효과 중 대부분은 실물 크기 모형과 미니어처 등을 적극 활용한 것으로, 조르주 멜리에스(Marie Georges Jean Méliès, 1861-1938)의 <달세계 여행(Le Voyage dans la Lune, 1902)>때부터 있어왔던 전통적 특수 효과(Special effects, SFX)의 범주에 속한다.³²⁾

<쥬라기 공원>이 공룡을 보여주는 방식 또한 연산량의 관점으로 새롭게 이해해볼 수 있다. 이 관점에서 보면, 공룡의 거대한 크기는 장애물이 아니라 오히려 장점이다. 가령 고층 빌딩만한 브라키오사우루스(Brachiosaurus)의 전체 모습을 한 화면에 담으려면 그것이 카메라로부터

32) 크리스티앙 메즈에 따르면, 전통적 특수 효과는 “평범한 촬영의 현장 작업에 반드시 추가적으로 특정 기술을 더해 얻어지는 효과”를 의미한다. 멜리에스의 <달세계 여행>에서는 주인공 일행이 달의 원주민과 교전할 때 그들이 연기처럼 사라지는 장면에서 사용된 기법이 이에 해당한다. 크리스티앙 메즈, 『영화의 의미작용에 관한 에세이 2』, 이수진 역, 문학과 지성사, 2011, 216쪽.

멀리 떨어진 곳에 있다는 식으로 장면을 연출해야 하는데, 이 거리라면 CG/CGI로 만든 공룡의 가죽을 굳이 세밀하고 정교한 수준까지 작업하지 않아도 된다. 떼 지어 이동하는 공룡 한 무리를 멀리에서 보여주는 장면도 같은 맥락에서 이해할 수 있다. 이 때 CG/CGI 객체의 크기는 앞에 비해 줄어들지만 수는 늘어나고 이동 속도도 더 빨라진다. 말하자면 다수의 소형 객체가 고속으로 이동하는 상황을 CG/CGI로 구현해야 하는 경우인데, 이 상황이라면 제한된 연산 성능을 각 객체의 정밀도 대신 전체 객체의 수를 늘리는 데에 집중하는 편이 더 합리적이다. 여기에 몸집이 클수록 중력의 영향을 더 많이 받아서 빠르게 움직이기 어려워진다는 기본적 과학 상식을 더하면 움직임도 최소화할 수 있다. 이는 작업량과 연산량을 줄이는 가장 직접적인 방식이자 공룡의 압도적인 거대함을 드러내 보이는 가장 효과적인 방식이기도 하다.

멀리 두어야 겨우 한 화면에 담길 정도로 크기가 크다는 말은 곧 가까운 거리에서는 공룡의 일부만 화면에 담긴다는 말이기도 하다. 이 경우라면 CG/CGI 대신 공룡의 각 부위를 정교하게 재현한 모형들을, 즉 CG/CGI 대신 전통적 SFX를 활용하는 편이 더 논리적이며 효율적이다. 카메라에 근접한 공룡의 CG/CGI작업이 불가피한 경우에도 대안은 있다. 모션 블러(Motion blur)는 빠른 속도로 움직이는 물체를 느린 셔터 스피드로 촬영할 때 그 속도 때문에 정확한 형태가 아니라 대략의 형태와 잔상만이 기록되는 광학적 현상이다. <쥬라기 공원>은 공룡이 카메라와 지근거리에서 빠르게 이동하는 장면에서 이 모션 블러 효과를 적극적으로 활용함으로써 CG/CGI 작업량을 덜 수 있었다.

<쥬라기 공원>과 같은 해에 공개된 펜티엄과 이것으로 대표되는 고성능의 대중화는 작업에 힘을 신기에 충분했다. 물론 펜티엄이 <쥬라기 공원>의 개봉보다 석 달 먼저 일반 대중에 공개된 것은 사실이지만, ILM이 <쥬라기 공원>의 작업에 펜티엄 시스템을 도입하여 활용했을 가능성은 매우 낮다. 펜티엄 출시 전 개발 단계에서부터 인텔과 ILM이 긴밀하

게 협업했을 가능성, 바꿔 말하자면 ILM이 ‘개발 단계 시제품’을 먼저 써 볼 수 있는 특혜를 누렸을 가능성을 완전히 배제할 수는 없으나 이 또한 현실성은 적다. 당시 ILM의 자체 개발 소프트웨어들의 대부분은 SGI 시스템용이었고, SGI 시스템과 펜티엄 시스템은 호환 가능(Compatible)한 부분이 거의 없다고 보아도 무방할 정도로 기본적 구조에서부터 큰 차이를 보인다. 간단히 말해서 SGI용으로 작성된 소프트웨어는 펜티엄에서 작동하지 않으며, 이것을 가능하게 만들려면 소프트웨어를 사실상 처음부터 다시 작성하는 수준의 대대적 수정이 필요하다. 이는 상식적이지도, 논리적이지도, 전략적이지도 않은 접근이다. 설령 ILM이 펜티엄 시스템용 소프트웨어를 몇 주 만에 다시 작성하는 데 성공했다고 하더라도, 90년대 초 VFX CG/CGI 작업에 걸리는 시간을 고려한다면 새 소프트웨어를 <쥬라기 공원>에 활용할 수 있었을 확률은 마찬가지로 0에 수렴한다.³³⁾

3. <공룡 대탐험(1999)>의 더 정교하고 세밀해진 공룡 표현

새 장비를 구비해야 하는 신생·후발 스튜디오들에게 SGI 시스템과 펜티엄 시스템의 구조적 차이는 걸림돌이 아니라 오히려 기회였다. 이들은 구하기 쉬워진 저비용·고성능 시스템을 등에 업고 실질적인 성과들을 내면서 업계 내에서 입지를 다지기 시작했다. 프레임스토어(Framestore)가 만들고 BBC가 1999년 10월 4일부터 11월 8일까지 총 6주에 걸쳐 방영한 자연 다큐멘터리 <공룡 대탐험 (Walking with Dinosaurs)>은 이러한 경향을 살펴보기에 적절한 사례다.

33) Lev Manovich, *The Language of New Media*, Cambridge, Massachusetts (United States): The MIT Press, 2001, pp.201~203.



[그림 4] <공룡 대탐험>의 CG/CGI³⁴⁾

공룡이라는 공통 소재를 제외하면 <공룡 대탐험>과 <쥬라기 공원>은 거의 모든 부분에서 차이를 보인다. 일단, 한 편당 30분씩 총 여섯 편 180분을 전량 CG/CGI로 작업한 Full-CG/CGI 다큐멘터리였다. 앞서 살펴본 바와 같이, <쥬라기 공원>의 CG/CGI 분량은 다 해서 6분 남짓에 불과하며 그마저도 작업에 사용한 시스템의 연산 성능 한계로 인해 CG/CGI 객체-가상 카메라간 거리 유지 등 각종 우회책을 동원해야만 했다. 반면 <공룡 대탐험>은 <쥬라기 공원>에 비해 CG/CGI 객체와 가상 카메라 사이의 평균 거리가 더 가깝다. 즉 <공룡 대탐험>은 <쥬라기 공원>보다 공룡을 더 크고 더 가깝게 보여준다. 이는 화면이 충분한 설득력을 가지려면 각 CG/CGI 객체의 묘사가 더 세부적인 부분까지 더 정교하게 이루어져야 함을, 즉 작업량과 연산량이 증가해야 함을 의미한다. CG/CGI 객체의 묘사 수준에는 공룡뿐만 아니라 일부 환경 객체들, 즉 CG로 구현한 나무, 풀, 호수, 구름 등도 포함된다.

34) Tim Haines, *Walking with Dinosaurs*, 1999.

요컨대 <공룡 대탐험>은 공룡을 비롯하여 그 공룡들이 살아있던 당시 지구의 전반적인 모습을 CG/CGI를 이용하여 설득력 있는 수준까지 구현해 냈으며, 더 다채로운 공룡의 생태를 더 가까이에서, 더 생생하게, 더 많이, 그리고 더 오래 보여주는 데에 성공했다는 점에서 특기할 만하다. 여기에 이 프로젝트가 첫 기획 회의부터 최종 후반작업(Post-production)까지 3년도 채 걸리지 않았을 정도로 짧은 기간 안에 이루어졌다는 사실을 함께 고려한다면,³⁵⁾ 고성능의 대중화로 인한 기술 환경의 변화가 모든 산업에서 가장 값진 비용 중 하나인 시간을 절감하는 데에 특히 주효했음을 파악할 수 있다.³⁶⁾

이 시기 정보기술 분야의 발전은 독일의 매체 철학자 프리드리히 키틀러(Friedrich Kittler, 1943-2011)와 그가 인용한 토머스 핀천(Thomas Pynchon, 1937-현재)이 언급한 “계단식 함수(Step-function)”³⁷⁾ 형태의 발전 양상을 떠올리게 한다.³⁸⁾ 계단식 함수는 초반의 급격한 상승세가 오래 지속되지 않고 비교적 짧은 시간 안에 정체기(Plateau)에 접어드는 로그 함수의 양상과, 정체기가 비교적 오래 지속되다가 어느 지점에서 급격하게 상승 즉 도약(跳躍, Leap)³⁹⁾하는 지수 함수의 양상이 하나의

35) 참고. Mark Walton, 앞의 기사.

36) <공룡 대탐험>의 총 제작비는 약 600만 파운드(약 800만 달러)로, <쥬라기 공원>의 6300만 달러에 비해 훨씬 낮다. 가장 단순하게 <쥬라기 공원>의 CG/CGI 장면과 비(非) CG/CGI 장면의 제작비와 완전히 동일하다고 가정해도, 이 영화의 총 CG/CGI 비용은 6300만 달러의 4.7퍼센트인 약 300만 달러, 그리고 1분당 CG/CGI 비용은 약 50만 달러에 이른다는 계산이 나온다. 인플레이션까지 고려하면 <공룡 대탐험>과 <쥬라기 공원>의 비용 차이는 더 벌어진다. 1993년의 6300만 달러는 1999년의 7264만 달러에 상응하고, 93년에 1분당 약 50만 달러였던 <쥬라기 공원>의 CG/CGI 비용은 99년에는 약 57만 달러까지 상승한다. 그에 비해, <공룡 대탐험>의 CG/CGI 비용은 같은 방식으로 계산해도 1분당 4만5천 달러를 넘지 않는다. 이는 상업영화용 CG/CGI의 작업량이 TV 방송용의 그것에 비해 현저히 더 많다는 점을 감안하더라도 유의미한 차이이다.

37) Thomas Pynchon, *V.*, Philadelphia, Pennsylvania (United States): J. B. Lippincott & Co., 1963; reprinted in New York (United States): Harper & Row, 1990, p.331.

38) 프리드리히 키틀러, 『광학적 미디어: 1999년 베를린 강의 - 예술, 기술, 전쟁』, 윤원화 역, 현실문화, 2011, 184쪽.

39) 이 ‘도약 지점’은 레이 커즈와일(Raymond “Ray” Kurzweil) 등이 제시한 ‘기술적 특이

함수에서 번갈아 출현하는 모습을 계단에 비유한 표현이다.

이 관점에서 보면, 90년대 초중반에 이루어진 고성능의 대중화는 개인용·가정용 컴퓨터의 보급 이후 한 동안 정체기에 머무르던 정보기술 분야에 새로운 활력을 고취한 도약 지점으로 간주하기에 충분하다. 뿐만 아니라, 고성능의 대중화는 정보기술이 형성한 기술 환경 그리고 이 환경과 긴밀하게 상호작용하던 여러 산업 분야들에 일대 전환점을 제공했다는 점에서 중요한 의의를 갖는다. 무엇보다도, 높은 부동소수점 연산 성능의 대중화는 VFX CG/CGI 분야가 한 단계 도약하는 직접적인 계기가 되었다는 점에서 이 후에 잇따른 도약들의 촉발점으로 보기에 손색이 없다.

V. 맺는 말

지금까지, 우리 시대 정보기술 환경을 오늘날의 모습으로 형성한 핵심 요소 중 하나인 부동소수점 개념과 중요성을 동시대적 맥락에서 첫 번째 도약 지점이라고 볼 수 있을 만한 사건과 사례들을 중심으로 살펴보았다.

이것이 ‘첫 번째’인 이유는 부동소수점 연산 성능의 도약적 향상이 한번에 그치지 않았기 때문이다. 1990년대 초중반에 VFX CG/CGI 분야에서 잠재력을 인정받은 부동소수점 개념은 빠른 속도로 정보기술 전반에 통합되었다. 아무리 늦게 보아도 지난 세기가 끝나기 전에, 부동소수점 연산 성능은 이미 VFX CG/CGI 작업을 비롯한 컴퓨터 그래픽스 영역 일체, 나아가 제반 시각 데이터 처리(Visual data processing) 분야에서 가장 중요한 성능 지표(Index) 중 하나가 되었다. 이는 이들 작업에서 필수불가결한 하드웨어인 그래픽스 처리장치(Graphics processing unit, GPU) 개념의 등장과 발전이 그 자체로 부동소수점 연산 성능 향상의 역

점(Technological singularity)*으로도 이해할 수 있다. 참조. 레이 커즈와일, 『특이점이 온다: 기술이 인간을 초월하는 순간』, 남윤호 역, 김영사, 2007.

사와 다름없다는 점에서 특히 도드라진다.

높은 부동산소수점 연산 성능의 대중화와 GPU의 보급화는 VFX CG/CGI분야의 발전뿐만 아니라 <울펜슈타인 3D (Wolfenstein 3D, 1992)>, <둠 (Doom, 1993)>, <퀘이크 (Quake, 1996)> 등 3D 게임 콘텐츠들의 출현도 함께 견인했다. 그리고 이 시기 특히 많이 생겨난 이드 소프트웨어(id Software LLC) 등의 3D 게임 개발사들은 오늘날 VR(Virtual reality), AR(Augmented reality 또는 Alternative reality), MR(Mixed reality), XR(eXtended reality), 그리고 메타버스 등의 개발을 주도하고 있다.

세기 전환기 즈음 부동산소수점 연산 성능은 다시 한 번 도약했고, 시각 데이터 처리를 넘어 일반 데이터 처리(General data processing) 분야에까지 광범위하게 확대·적용되기 시작했다. CPU보다 GPU가 부동산소수점 연산에 더 적합한 하드웨어라는 점에 주목한 연구자들은 GPU를 재프로그래밍(Reprogram)함으로써 GPU의 부동산소수점 연산 성능을 일반 데이터 처리에 유용(流用, Reappropriate)할 방안, 즉 GPGPU(General-Purpose computing on GPU, GPU를 활용한 범용 연산)컴퓨팅을 적극적으로 모색했다. 이에 GPU 제조사들은 GPU의 프로그램성(Programmability)을 제고하는 것으로, 즉 더 유연하게 재프로그래밍 할 수 있는 GPU(Programmable GPU)를 출시하는 것으로 응답했다.

GPGPU 컴퓨팅은 사실상 CPU에만 의존해왔던 기존의 전통적 데이터 처리 및 연산 방식에 근본적인 변화를 견인함과 동시에 정보기술에 부동산소수점 개념을 더 광범위하게 통합함으로써 정보기술로 다룰 수 있는 데이터의 정밀도와 정확도를 완전히 새로운 수준으로 끌어올렸다. 물론 시각 데이터 처리가 주 목적인 GPU의 특성상 초기 GPGPU 연구들의 대부분은 시각 데이터에서 일반 데이터를 추출하거나 시각 데이터를 일반 데이터로 변환하는 것에 관련되었지만,⁴⁰⁾ 재프로그래밍을 통한 GPGPU의

40) 참고. James Fung & Steve Mann, “Using multiple graphics cards as a general purpose parallel computer: applications to computer vision” in *Proceedings of the 17th*

유연성과 다목적성은 동시대 인공지능 연구 분야에까지 영향을 주어 마침내 알렉스넷(AlexNet)이나 알파고(AlphaGo)로 대표되는 동시대적 의미의 심층 인공 신경망(Deep neural network)과 이에 기반을 둔 동시대 인공지능(Contemporary AI) 등의 성과들을 이끌어내기에 이르렀다.⁴¹⁾

우리 시대는 어제를 미처 다 파악하기도 전에 오늘을 맞이하고, 오늘이 왔음을 채 인지하기도 전에 내일에 자리를 내어줄 정도로 빠르게 흐른다. 그리고 이 흐름의 중심에는 정보기술이 있다. 요컨대 정보기술은 동시대 문화예술 콘텐츠 분야에서 가장 편재(偏在, Omnipresent)하는 요소 중 하나다.

표현 층위와 의미작용의 관계를 고려할 때, 오늘날의 영상을 분석함에 있어 영상기호학이 우리 시대 문화기술 환경의 맥락을 고려하지 않을 이 유는 적다. 현존하는 피사체를 포착하고 재현한 시각기호 및 이것의 의미작용이 ‘현존하지 않지만 마치 현존하는 것 같아 보이는’ CG/CGI 시각기호 및 그것의 의미작용과 같을 수는 없기 때문이다. 만일 전자에 천착하는 연구 분야를 영상기호학이라고 부를 수 있다면, 후자에 집중하는 분야는 디지털 영상기호학이라고 부를 수 있을 것이다. 그리고 전통적 영상 기술들에 관한 명확한 이해가 영상기호학에 동반되어야 함을 고려한다면, 디지털 영상기호학 또한 디지털 영상에 유관한 기술 즉 정보기술에 관한 정확한 이해가 수반되어야 함은 당연하다.

자연과학이 한 단계 도약하던 지난 세기, 기호학은 자연과학의 객관성과 체계성을 비평과 분석에 도입함으로써 문화예술 콘텐츠 연구의 새로

International Conference on Pattern Recognition (ICPR) 2004 Volume 1, <https://doi.org/10.1109/ICPR.2004.1334339>, Cambridge (United Kingdom), 2004, pp.805~808.

- 41) 통상 기계 학습 혹은 머신 러닝(Machine learning)으로 일컬어지는 인공지능의 학습과 훈련에서 가장 주요한 변수는 주어진 정보기술 하드웨어의 성능, 그 중에서도 GPU의 부동소수점 연산 성능을 활용한 GPGPU 성능이다. 이는 인공지능 연구에 필요한 하드웨어의 성능을 측정하는 단위가 초당 부동소수점 연산량을 의미하는 플로팅스(Floating point operations per second, FLOPS)라는 점에서 분명해진다. GPGPU 개념과 동시대 인공지능 연구의 관계에 관한 논의는 후속 연구에서 계속할 수 있기를 기대한다.

운 장을 열었다. 이번 세기, 학계는 예술과 기술을 함께 다루는 융합 담론들의 절대량 부족과 실질성 결핍이라는 과제에 당면한 듯 보인다. 융합 연구, 학제 간 연구 등은 더 늦기 전에 이러한 시대의 요구에 부응하기 위한 학계의 시도에 다름 아니다. 디지털 영상을 비롯한 문화예술 콘텐츠의 개발자, 창작자, 소비자, 그리고 기호학자 모두가 공유할 수 있는 공통 기반의 수립이 필요한 때다.

참고문헌

- 이수진, 「현대 사이언스 픽션 영화의 기호학적 읽기: <터미네이터>의 테크놀로지에 관한 시각적 재현을 중심으로」, 『기호학연구』 44집, 한국기호학회, 2015, pp.203~225.
- 크리스티앙 메츠, 『영화의 의미작용에 관한 에세이 2』, 이수진 역, 문학과 지성사, 2011.
- 프리드리히 Kittler, 『광학적 미디어: 1999년 베를린 강의 - 예술, 기술, 전쟁』, 윤원화 역, 현실문화, 2011.
- André Leroi-Gourhan, André, *Le geste et la parole : la mémoire et les rythmes*, Paris: Éditions Albin Michel, 1965.
- Bernard Stiegler, *La technique et le temps, Tome 2 : La Désorientation*, Paris: Éditions Galilée, 1996.
- Christian Metz, *L'Énonciation impersonnelle, ou le site du film*, Paris: Méridiens Klincksieck, 1991.
- Friedrich A. Kittler, "There Is No Software" in *Stanford Literature Review* 9:1 (Spring 1992), pp.81-90.
- _____, *Optische Medien: Berliner Vorlesung 1999*, Berlin: Merve, 2002.
- IEEE, "IEEE Standard for Floating-Point Arithmetic" in *IEEE Standard 754-2019 (Revision of IEEE 754-2008)*, <https://doi.org/10.1109/IEEESTD.2019.8766229>, July 2019.
- Jack Burnham (Ed.), "Notes on art and information processing", in *Software* (exhibition catalogue), New York: Jewish Museum, 1970.
- John F. Hughes et al., *Computer Graphics: Principles and Practice*, Boston: Addison-Wesley, 2013 (for 3rd edition).
- Lev Manovich, *The Language of New Media*, Cambridge: The MIT Press, 2001.
- _____, *Software Takes Command*, New York: Bloomsbury Academic, 2013.
- Thomas S. Kuhn, *The Structure of Scientific Revolutions*, Chicago: The University of Chicago, 1962; 1970 (for 2nd edition).
- Vilém Flusser, *Für eine Philosophie der Fotografie*, Göttingen: European Photography, 1983.

인터넷 검색 자료

Mark Walton, “From Paintbox to PC: How London became the home of Hollywood VFX”, *Ars Technica*, 2016, <https://arstechnica.com/gaming/2016/05/how-london-became-the-home-of-hollywood-vfx>

Randall Hand, “What Led to the Fall of SGI”, *VizWorld*, 2009 April-May, <https://vizworld.com/tag/sgi-bts>

기타 자료

James Cameron, *Terminator 2: Judgment Day*, 1991.

Steven Spielberg, Steven, *Jurassic Park*, 1993.

Tim Haines, *Walking with Dinosaurs* (TV Documentary), 1999.

An Interdisciplinary Study on the Correlation Between FLOPS Performance and VFX CG/CGI Workflow Improvements: FLOPS Improvements in the Early 90s and Examples of CG/CGI VFX

Yoon, Na-Ra

This article aims to ponder the lack of competitiveness and the impracticality of interdisciplinary research on art as well as on information technologies. The concept of floating point system and FLOPS is important, because all VFX CG/CGI workflow except its surface or “most superficial layer” is fundamentally writing mathematical formulas and manipulating mathematical data. Although the floating point system offers more precision and accuracy, it requires more computing power than the fixed point system. In the early 1990s, low-cost and high-performance general-purpose hardware such as Intel’s PentiumTM processor offered significantly more FP performance or “FLOPS” than before. Such a leap in performance led not only to the advent of a renewed technological environment, but also to the sophistication of VFX CG/CGI workflows, and ultimately, to the declination of a company such as SGI. This article is an experimental attempt that tries to re-examine the then situation not through another traditional point of view, but through a contemporary interdisciplinary one.

Keywords : Information technologie (IT), Floating point (FP), FLOPS, Computer graphics (CG), Computer-generated imagery (CGI), Art & Technologie (Art & Tech), Interdisciplinary research

투고일: 2022. 03. 27./ 심사일: 2022. 04. 14./ 심사완료일: 2022. 04. 14.