

케이지 기반의 대규모 운동 경로 편집
(Cage-based Large-scale Motion Path
Editing)

지도교수 : 이제희

이 논문을 공학학사 학위 논문으로 제출함.

2011 년 12 월 22 일

서울대학교 공과대학

컴 퓨 터 공 학 부

고 우 중

2012 년 2 월

케이지 기반의 대규모 운동 경로 편집
(Cage-based Large-scale Motion Path
Editing)

고우종

서울대학교 컴퓨터공학부

wjngkoh@gmail.com

2011 년 12 월 22 일

차 례

제 1 장	서론	3
제 2 장	관련 연구	6
제 1 절	Mean Value Coordinates	6
제 2 절	Green Coordinates	7
제 3 장	케이지 기반의 운동 경로 편집	8
제 1 절	개괄	8
제 2 절	알고리즘	9
제 4 장	결과	10
제 5 장	결론	13

그림 차례

1.1	여러 운동 경로로 구성된 장면	3
1.2	그림 1.1 에 2차원 삼각형 메시 (Delaunay) 를 씌운 모습 . . .	5
4.1	변형 전	11
4.2	변형 후 (좌) 본 논문의 방법 (우) 기존 Laplacian editing . .	11

제 1 장

서론

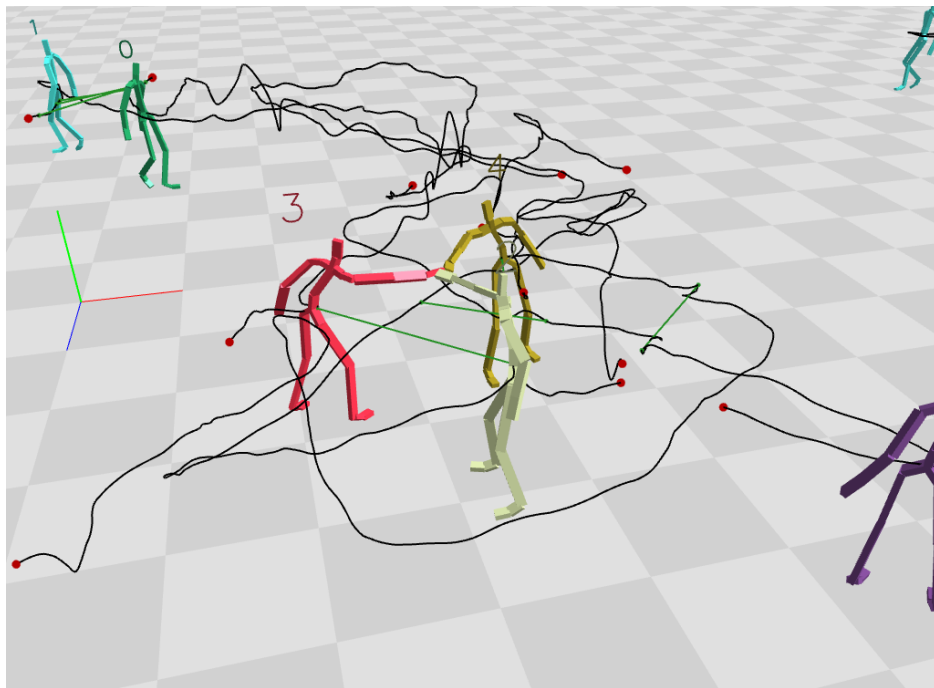


그림 1.1: 여러 운동 경로로 구성된 장면

최근 들어 매우 큰 규모의 군중이 등장하는 장면이 영화나 애니메이션, 비디오 게임 등에 자주 등장하고 있다. 이러한 군중 장면들은 보통 그

림 1.1처럼 등장인물들끼리 협동 혹은 적대하는, 여러 개의 상호작용들로 구성되어 있다. 예를 들면 여러 명의 집꾼이 같은 물건을 동시에 들어서 옮기는 경우나 아니면 여러 명의 병사들이 서로 전투를 하고 있는 장면을 들 수 있다. 이러한 상호작용이 운동 경로 (motion path)에 있어서는 일종의 제약조건 (constraint)으로 작용하는데, 그 이유는 바로 상호작용에 참가하는 각 캐릭터들이 해당 상호작용에 참여하는 다른 캐릭터들과 상대적인 위치, 방향, 타이밍 면에서 잘 조정되어야 하기 때문이다. 본 논문의 목표는 이러한 복잡하게 서로 의존하는 운동 경로들을 상호작용들은 최대한 유지하면서 사용자가 원하는 대로 실시간으로 편집하는 새로운 방법을 개발하는 것이다.

이 문제의 가장 근본적인 어려움은 바로 두 요구사항, 즉 속도와 정확도가 서로 상충하는 역의 관계에 있다는 것이다. 우선, 앞서 언급한 이 제약조건들을 모두 동시에 만족시키기 위해서는 운동 경로 편집 시 매우 정확하고 총체적인 계산이 필요하다. 기존의 방법은 [5] 운동 경로들의 모든 프레임을 하나의 선형 시스템에 포함시킨 후 풀으로써 이 문제를 해결하였다. 하지만 이는 질적인 품질을 높이는 대신, 계산 복잡도를 프레임의 총 개수에 의존하게 만듦으로써 기존의 방법을 확장 가능 (scalable)하지 않게 만들었다. 다시 말해 프레임의 총 개수가 증가하게 되면 기존의 방법은 인터랙티브한 편집 속도를 달성하지 못하게 된다.

본 논문의 가장 핵심적인 아이디어는 바로 이처럼 복잡하게 상호 종속적인 운동 경로들의 집합은 서로 분리된 곡선들처럼 따로따로 변형되기보다는 마치 2차원 삼각형 메시 (mesh)처럼 같이 변형된다는 관찰에 기반한다. 다시 말해 본 논문에서는 케이지 기반의 운동 경로 편집을 이 문제의 해결책으로 제시한다. 이를 간단히 설명하면, 먼저 운동 경로들을 모두 덮도록 오목 꺾데기를 만들고 이를 작은 삼각형들로 나뉘서 그림 1.2처럼

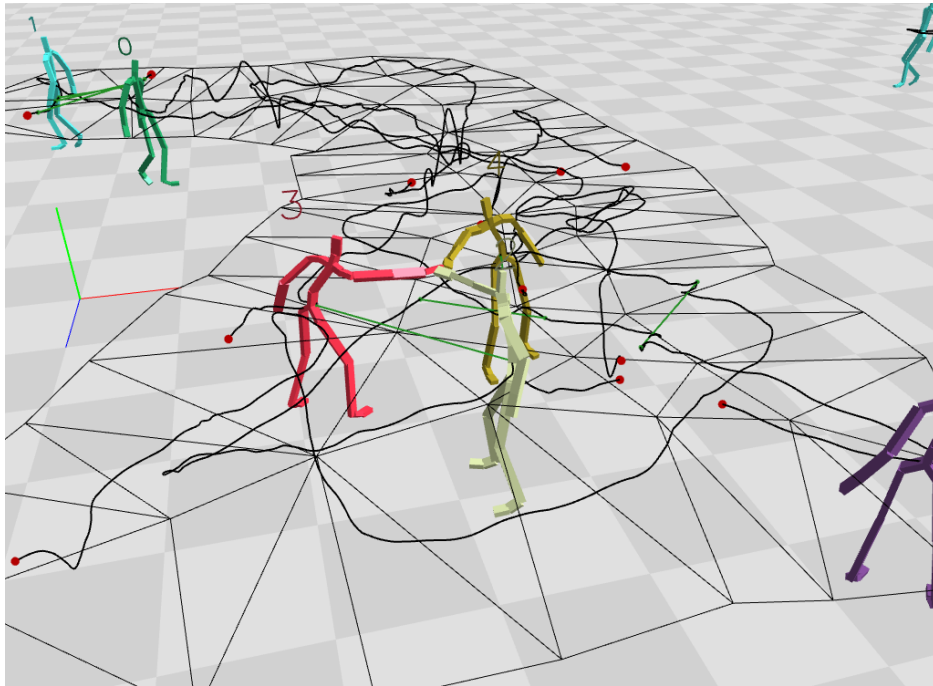


그림 1.2: 그림 1.1에 2차원 삼각형 메시 (Delaunay)를 씌운 모습

삼각형 메시지를 만든다. 그리고 운동 경로들의 모든 프레임의 일반화된 무게중심좌표 (중간값 좌표, 그린 좌표 등) 를 이 삼각형 메시의 외곽 꼭지점들에 대하여 계산한다. 그러고 나면 이 케이지를 가능한 한 잘 휘지 않게 (as-rigid-as-possible) 변형함으로써 그에 맞춰 변형된 운동 경로들의 프레임을 계산해낼 수 있다. 정리하면 변형된 프레임들의 좌표는 프레임들의 초기 중간값 좌표 (가중치)와 변형된 케이지의 외곽 꼭지점들 좌표의 가중합 (weighted sum) 이다.

제 2 장

관련 연구

기존의 방법 [5]은 Laplacian curve editing [7], 더 정확히 말하면 Laplacian에 비율 조정 (scale adjustment)을 추가한 as-rigid-as-possible curve editing [4]을 응용하였다. 이를 이용해 운동 경로를 모든 제약조건들—사용자 정의 제약조건이나 상호작용에 있어서 캐릭터의 위치, 방향, 타이밍 등—을 만족시키면서 부드럽게 변형되도록 했다.

기존의 방법과 달리 새 케이지 기반의 운동 경로 편집은 중간값 좌표와 케이지를 도입하여, 운동 경로를 직접적으로 변형하지 않고 대신 케이지를 as-rigid-as-possible mesh editing [4]을 사용하여 변형하도록 하고 이로부터 운동 경로를 재구축하도록 하였다.

제 1 절 Mean Value Coordinates

본 논문에서는 무게중심좌표 (barycentric coordinates)의 일반화 중 하나인 중간값 좌표 (mean value coordinates)를 사용하였다. 이 좌표는 조화 함수를 위한 중간값 정리에서 착안하여 만들어진 좌표로서, 다각형의 꼭지

점들에 매겨진 값들로부터 중간값들을 보간하는데 유용하다 [1].

무계중심좌표는 볼록다각형에는 적용할 수 있지만 볼록하지 않은 다각형에 대해서는 일반적으로 적용할 수 없다. 반면에 중간값 좌표는 모서리끼리 스스로 교차하지 않는, 임의의 평면다각형에 대해서도 잘 정의된다 [2]. 운동 경로들을 정교하게 덮기 위해서는 오목 꺾대기 (concave hull)를 사용해야 하므로 본 논문에는 무계중심좌표보다 중간값 좌표가 더 적합하다.

중간값 좌표의 정의를 간단히 설명하면 다음과 같다. 2차원 상의 점 $x \in \mathbb{R}^2$ 와 순환적으로 정의된 단순 다각형 (simple polygon) $[v_0, v_1, \dots, v_k, v_{k+1} = v_0]$ 이 있다 하자. 그리고 각 α_i 를 그 다각형의 각 $\angle v_{i+1}, x, v_i$ 라 하자.

$$\lambda_i = \frac{w_i}{\sum_{j=1}^k w_j}, \quad w_i = 2 \frac{\tan(\alpha_{i-1}/2) + \tan(\alpha_i/2)}{\|x - v_i\|} \quad (2.1)$$

그러면 가중치 $(\lambda_0, \dots, \lambda_k)$ 가 바로 x 의 v_0, \dots, v_k 에 대한 중간값 좌표가 된다.

제 2 절 Green Coordinates

그린 좌표 (Green coordinates) 역시 오목다각형에도 적용할 수 있는 무계중심좌표의 일반화 중 하나이다. 계산 과정은 조금 더 복잡하지만, 그린 좌표는 2차원에서 순수 등각사상 (pure conformal mapping) 이므로 원래 운동 경로의 형태를 중간값 좌표보다 더 잘 보존하는 특성을 가지고 있다 [6].

제 3 장

케이지 기반의 운동 경로 편집

제 1 절 개괄

케이지 기반의 운동 경로 편집을 대략적으로 설명하면 다음과 같다. 먼저, 운동 경로들을 덮는 최소한의 볼록 혹은 오목 꺾데기를 만들고 이를 삼각형으로 나누어서 2차원 삼각형 메시를 구성한다. 그리고 as-rigid-as-possible triangle mesh editing [4]을 사용하여 이 삼각형 메시를 사용자가 원하는 대로 변형시킨다. 사용자는 이 삼각형 메시의 어떤 꼭지점이라도 마음대로 움직이고 고정시킬 수 있다. 그리고 나면 이 변형된 삼각형 메시, 정확히는 메시의 외곽 꼭지점들로부터 변형된 운동 경로들을 계산해 낼 수 있다.

더 자세히 설명하면, 운동 경로들의 모든 프레임은 이 삼각형 메시의 외곽 꼭지점들과 그 꼭지점들에 대한 중간값 좌표로 표현할 수 있다. 따라서 케이지의 역할을 하는 이 삼각형 메시가 원래 형태를 최대한 유지하려고 노력하면서 변형된다면, 중간값 좌표의 보간 (interpolation), 매끄러움 (smoothness), 그리고 선형 정확도 (linear precision) 성질 덕분에 중간값

좌표로 표현된 운동 경로들 역시 부드럽게 변형된다. 새 방법은 기존의 방법과는 달리 모든 프레임이 포함된 선형 시스템을 풀 필요가 없으므로 매우 빠른 속도로 계산을 완료할 수 있다.

제 2 절 알고리즘

앞서 언급했듯이 기존의 편집 방법 [5]은 *as-rigid-as-possible curve editing* [4]에 사용자 정의 제약조건을 추가한 다음 운동 경로에 적용함으로써 원하는 결과를 얻어내었다. 본 논문 역시 비슷하게 *as-rigid-as-possible triangle mesh editing* [4]에 사용자 정의 제약조건을 추가했지만, 이를 운동 경로가 아닌 케이지에 적용하였다. 그리고 이 변형된 케이지에서 중간값 좌표를 통해 운동 경로를 재구축함으로써 원하는 결과를 얻었다.

이 사용자 정의 제약조건을 추가하는 방법에 대한 실마리는 Takeo Igarashi의 2009년 후속 논문에서 발견할 수 있었다. 이 후속 논문의 5절, *Allowing Handles on Arbitrary Positions in the Mesh* [3, p.26]에 기술된 대로 구현하면 삼각형 메시의 꼭지점들뿐만 아니라 메시 안의 임의의 점들도 핸들로 잡고 고정시킬 수 있다. 이를 이용하여 운동 경로의 위치 제약조건을 구현할 수 있었고 사용자 정의 제약조건 역시 사용자가 메시의 꼭지점 뿐만 아니라 임의의 점을 잡고 변형할 수도 있게 되었다.

따라서 첫 번째 과정인, 제약조건들을 만족하면서 *as-rigid-as-possible*하게 삼각형 메시를 마음대로 변형하는 것은 비교적 수월하게 구현할 수 있었다. 그리고 두 번째 과정인, 변형된 메시에서 변형된 운동 경로들을 재구축하는 것은 프레임들의 초기 중간값 좌표와 변형된 메시의 외곽 꼭지점 좌표들의 단순한 선형 조합 (linear combination)이므로 벡터와 벡터의 곱으로 간단하게 구할 수 있었다.

제 4 장

결과

계산 소요시간 측정 실험에 사용된 환경은 다음과 같다. 테스트 머신의 CPU는 Intel Core2 Quad 2.40GHz 였으며 RAM은 4GB 였다. 운영체제는 Microsoft Windows 7 Enterprise (64-bit)를 사용하였고 실험 프로그램은 C++로 작성하여 Microsoft Visual Studio 2010 Professional로 컴파일하였다.

실험에 사용된 데이터는 그림 1.1과 그림 4.1에서 볼 수 있는, 모션 캡처된 운동 경로 6개로 구성된 하나의 장면이다. 모션 캡처 데이터는 16개의 카메라를 쓰는 Vicon 모션 캡처 시스템을 사용하여 120 프레임 / 초의 속도로 캡처하였고 30 프레임 / 초로 서브샘플링하였다.

항목	개수
총 운동 경로	6
총 프레임	3809
메시의 삼각형	110
메시의 꼭지점	81

표 4.1: 실험에 사용된 데이터

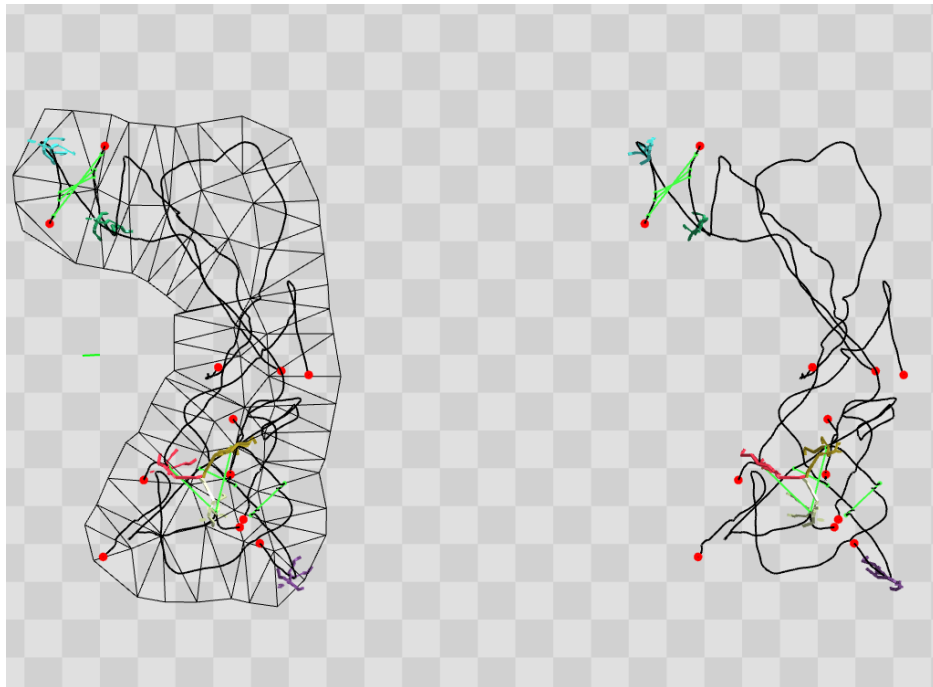


그림 4.1: 변형 전

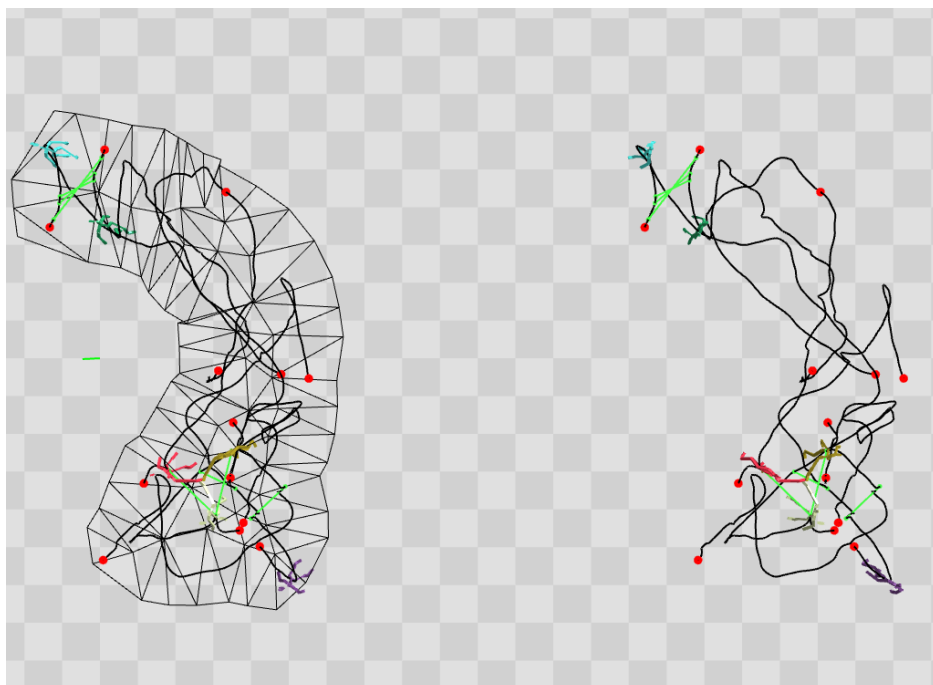


그림 4.2: 변형 후 (좌) 본 논문의 방법 (우) 기존 Laplacian editing

회차	계산 소요시간 (ms)		속도 향상 (배)
	기존 방법	새로운 방법	
1	1437	14	102.64
2	1469	14	104.93
3	1446	14	103.29
4	1391	13	107.00
5	1440	13	110.77
6	1444	13	111.08
7	1386	13	106.62
8	1444	13	111.08
9	1394	13	107.23
10	1442	13	110.92
평균	1429.3	13.3	107.55

표 4.2: 계산 소요시간 측정 결과

측정해본 결과, 표 4.2에서 볼 수 있듯이 기존 방법과 비교해 약 107.55 배의 속도 향상을 확인할 수 있었다.

제 5 장

결론

결과적으로 이 새로운 케이지 기반의 운동 경로 편집을 통해 계산 복잡도를 매우 큰 폭으로 낮출 수 있었고 편집 과정을 확장 가능 (scalable) 하게 만들 수 있었다. 그 이유는 이 새로운 방법의 계산 복잡도는 케이지라는 중간 매개체가 도입됨으로써 운동 경로들의 프레임 총 개수에 의존적이지 않고 케이지의 꼭지점 총 개수에 의존적이게 되었기 때문이다. 또한, 케이지의 꼭지점들을 새로운 조작 핸들로서 제시하여 애니메이터들이 거대한 군중 장면을 쉽게 편집할 수 있도록 하였다.

하지만 본 논문의 방법은 상대적 제약조건 (relative constraint)—상호 작용시 캐릭터 간의 거리, 방향 그리고 타이밍 등—이 작은 변형 (small deformation)의 경우에는 잘 맞지만 큰 변형 (large deformation)에서는 어긋날 수 있다. 따라서 차후 과제로는 상대적 제약조건을 확실하게 보장 해주도록 개선하는 것을 들 수 있다. 다시 말해 2차원 삼각형 메시 변형 시 삼각형의 형태와 사용자 정의 제약조건만 최대한 유지하도록 하는 것이 아니라 이 상대적 제약조건도 최대한 지키도록 만드는 것이 앞으로 남은 과제이다.

참고 문헌

- [1] M. S. Floater, "Mean value coordinates," *Computer Aided Geometric Design*, vol. 20, no. 1, pp. 19–27, 2003, ISSN: 0167-8396. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0167839603000025> (cit. on p. 7).
- [2] K. Hormann and M. S. Floater, "Mean value coordinates for arbitrary planar polygons," *ACM Trans. Graph.*, vol. 25, pp. 1424–1441, 4 Oct. 2006, ISSN: 0730-0301. [Online]. Available: <http://doi.acm.org/10.1145/1183287.1183295> (cit. on p. 7).
- [3] T. Igarashi and Y. Igarashi, "Implementing as-rigid-as-possible shape manipulation and surface flattening," *Journal of Graphics, GPU, and Game Tools*, vol. 14, no. 1, pp. 17–30, 2009. eprint: <http://www.tandfonline.com/doi/pdf/10.1080/2151237X.2009.10129273>. [Online]. Available: <http://www.tandfonline.com/doi/abs/10.1080/2151237X.2009.10129273> (cit. on p. 9).
- [4] T. Igarashi, T. Moscovich, and J. F. Hughes, "As-rigid-as-possible shape manipulation," *ACM Trans. Graph.*, vol. 24, pp. 1134–1141, 3

- Jul. 2005, issn: 0730-0301. [Online]. Available: <http://doi.acm.org/10.1145/1073204.1073323> (cit. on pp. 6, 8, 9).
- [5] M. Kim, K. Hyun, J. Kim, and J. Lee, "Synchronized multi-character motion editing," *ACM Trans. Graph.*, vol. 28, 79:1–79:9, 3 Jul. 2009, issn: 0730-0301. [Online]. Available: <http://doi.acm.org/10.1145/1531326.1531385> (cit. on pp. 4, 6, 9).
- [6] Y. Lipman, D. Levin, and D. Cohen-Or, "Green coordinates," *ACM Trans. Graph.*, vol. 27, 78:1–78:10, 3 Aug. 2008, issn: 0730-0301. [Online]. Available: <http://doi.acm.org/10.1145/1360612.1360677> (cit. on p. 7).
- [7] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel, "Laplacian surface editing," in *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, ser. SGP '04, Nice, France: ACM, 2004, pp. 175–184, isbn: 3-905673-13-4. [Online]. Available: <http://doi.acm.org/10.1145/1057432.1057456> (cit. on p. 6).