

# **mplayer – real time strategy: битва за улучшения видеоряда**

крис касперски aka мышъх, aka nezumi, aka souriz, aka elraton, no-email

**mplayer** – популярный кросс платформенный видеоплеер с огромным количеством функций и поистине безграничными возможностями по исправлению дефектов мастеринга DVD (и кривых рипов, выкаченных из сети), которые до этого приходилось преодолевать off-line перекодировкой в редакторах нелинейного видео-монтажа. час фиксим баги, четыре часа перекодируем, после чего часа два смотрим фильм (если он стоит того). а mplayer позволяет делать это в режиме реальном времени без отрыва рук от производства, тыфа, от клавиатуры.

## **введение**

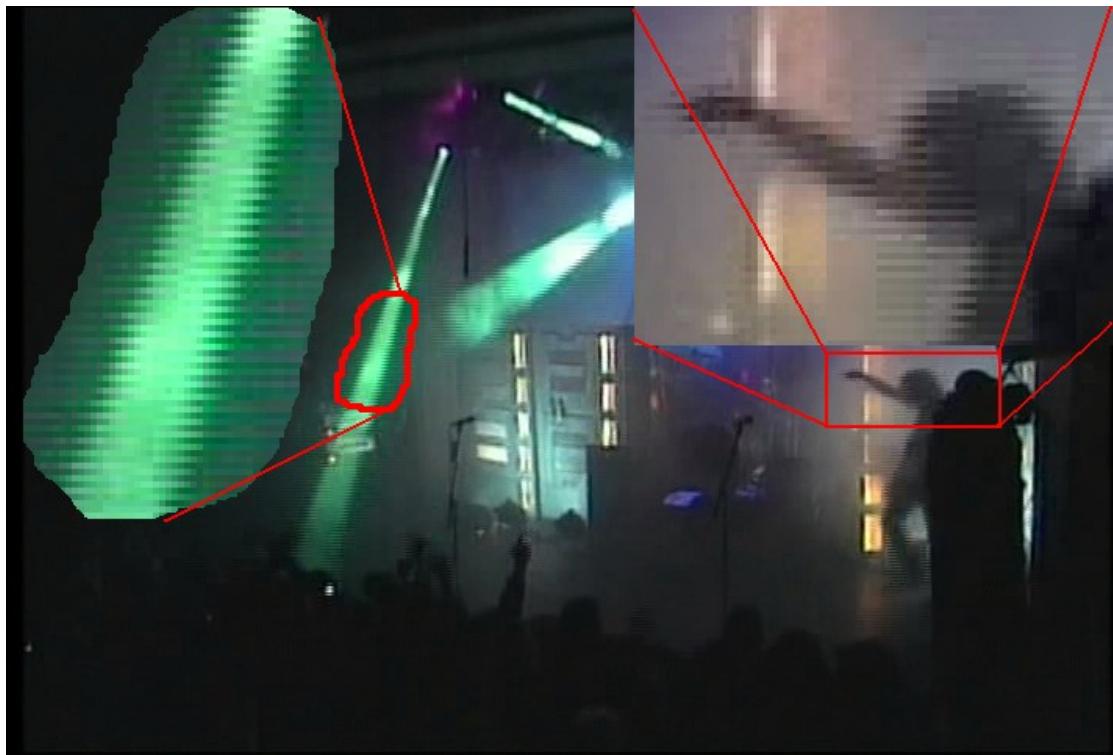
"]][акер" уже писал как собрать и обустроить mplayer (см. статью за февраль 2005 года), так что будем считать, что читатель, освоившийся с командной строкой (или с одной из многочисленных графических "морд") теперь хочет крови и зрелиц! В смысле зрелищного качества изображения, сражение за которое превращается в настоящую стратегию (в случае с mplayer'ом происходящую в реальном времени). Сколько крови ты готов пролить, читая многочисленные стандартны, мануалы, ковыряя исходники и пронираясь сквозь архивы рассылок для разработчиков плеера и входящих в его состав кодеков?

Победителя ждет солидный приз — реальное улучшение качества фильма/клипа, которое не обеспечивает никакой другой плеер с автоматической "коробкой передач". Нас окружает огромное количество отстойных DVD (и файлов, вытащенных из сети), которые виндузятники отправляют в морг не раздумывая, ну или сидят, смотрят и мучаются. Но мы — линуксоиды — имеем в своем арсенале мощное оружие, способное устраниить и противный interlacing, и мерзкий telecine и прочие гадости, способное испортить все впечатление от фильма/клипа.

Сражение за качество начинается!!!

## ***interlacing и его последствия***

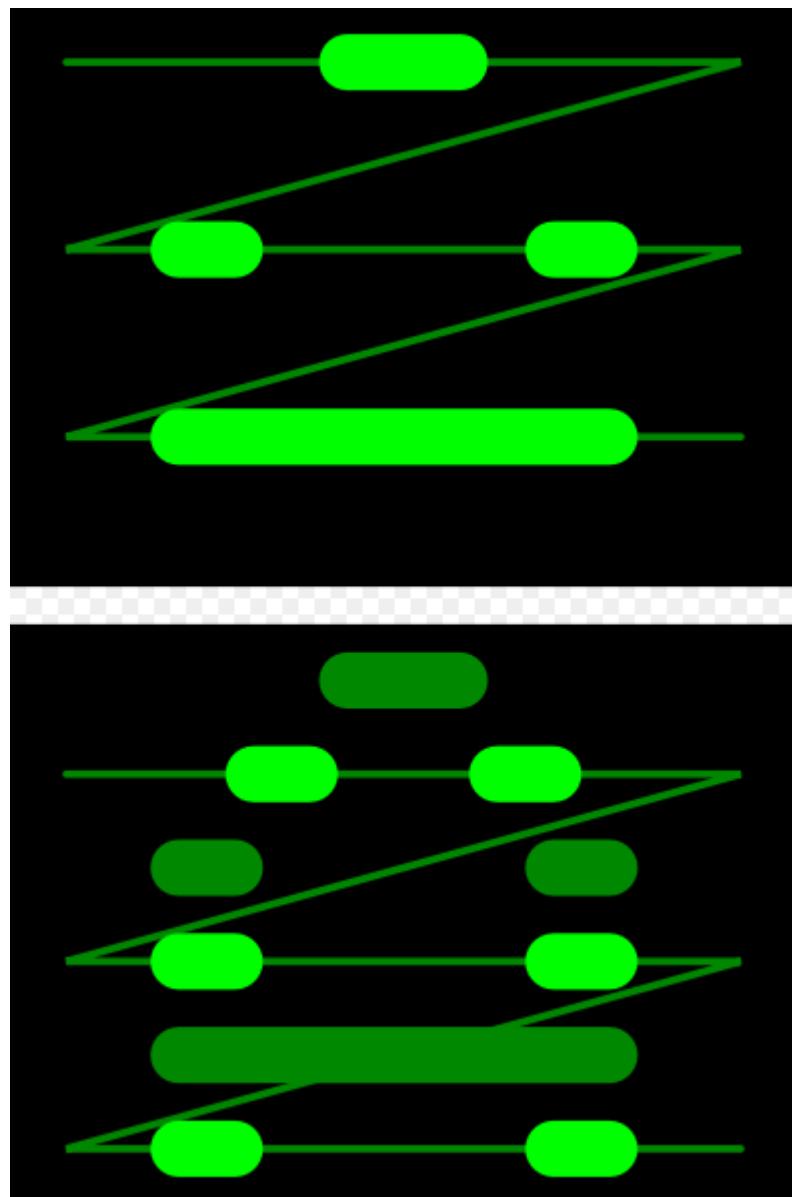
Берем в лапы DVD-диск "PAIN – live is overrated", вставляем его в привод и видим... что он записан в через-строчечном (ч/с) режиме (по-английски \_interlaced\_), а просмотр ч/с видеоматериала на устройстве с прогрессивным отображением (progressive или non-interlaced), к которым относятся многие современные телевизоры и все без исключения мониторы, оставляет жутковатое впечатление, высаживающее на измену и полный негатив.



**Рисунок 1 просмотр ч/с видео на прогрессивном мониторе**

Подробное объяснение этого термина легко найти на Википедии (<http://en.wikipedia.org/wiki/Interlaced> и <http://de.wikipedia.org/wiki/Deinterlacing>), поэтому мышьхъ будет предельно краток. Возьмем ч/с камеру системы PAL с заявленной частотой 25 кадров/сек. На самом деле никаких кадров (frames) в камере нет, зато есть поля или полукадры (fields) и там их 50 штук в сек. Наводим камеру на мяч нажмем "REC". За короткий промежуток времени (~1/200 сек) камера сканирует четные строки и записывает их в первый полукадр. После чего ждет 1/50 сек. и сканирует нечетные строки, записывая их в следующий полукадр. И так до тех пор, пока не надоест.

Полукадры выводятся на устройство ч/с отображения в том же порядке, в котором снимались. Сначала электронный луч прорисовывает четные строки первого полукадра, а через 1/50 сек — нечетные. За это время четные строки успевают поблекнуть, снижая четкость изображения.



**Рисунок 2 ход электронного луча при выводе буквы "A" в ч/с режиме**

Монитор (устройство прогрессивного отображения) — совсем другое дело. Два полукадра объединяются и выводятся за один проход, что существенно повышает четкость. Но... вот по мячу пнули и он полетел с огромной скоростью. За 1/50 сек. мяч сместится на расстояние сопоставимое со своим диаметром и при объединении полукадров на прогрессивном устройстве мы увидим два мяча, расчерченных полосами фона, образующими характерную "гребенку", смотреть на которую без содрогания невозможно. На ч/с устройствах отображения объединения соседних кадров не происходит и мы видим две фазы движения мяча, отделенные друг от друга 1/50 сек, но увы, перевести монитор в ч/с режим невозможно.



**Рисунок 3 так выглядит неподвижный (слева) и движущийся (справа) мяч, снятый в ч/с режиме и выводимый на прогрессивный монитор**

Но два мяча — это ерунда. Мы же не футбольный матч смотрим, а концерт группы Pain с кратковременными вспышками ослепительного света. В один полукадр попадает красная (ну, например), в следующий — фиолетовая. Совмещение двух разных цветов в одном кадре приводит к... гм... некоторому подобию первой детской неожиданности. А при резком смене сцены что происходит? Правильно! С вероятностью 50 на 50 полукадр сцены A смешивается с полукадром сцены B и мы видим shit.



**Рисунок 4** вспышки света выглядят просто отвратительно

Вот и приходится прибегать к различным deinterlace-алгоритмам, которые делятся на плохие, очень плохие и совсем отстойные. (см. "How and why every single deinterlacer sucks"/"как и почему все deinterlacer'ы сосут" — <http://lists.mplayerhq.hu/pipermail/mplayer-docs/2005-March/004815.html>).



Рисунок 5 "призрачное" изображение при резком смене сцены

Почему так мы узнаем чуть позже, пока же отметим, что возможность выбора произвольного фильтра в mplayer'e позволяет добиться максимально возможного качества. Остальные плееры либо вообще не поддерживают таких фильтров, либо выбирают их на автомате, что иногда (иногда!) обеспечивает вполне приемлемое качество, но чаще ухудшает его. Почему?! А вот почему!!!

Самое простое, что можно сделать для ликвидации "гребенки" — это выкинуть четные (или нечетные) поля, а оставшиеся растянуть по вертикали для сохранения оригинального аспекта ("vf field=0"). Мы потеряем 50% вертикального разрешения и половину фаз движения. Однако, во многих случаях, это наименьшее зло, чем артефакты продвинутых фильтров (см. рис. 6), которых достаточно много, например, линейная (linear) или кубическая (cubic) интерполяции ("vf pp=li" и "-vf pp=ci" соответственно). Только это ничем не лучше "-vf field=0". Нечетные строки (принадлежащие нечетным полукадрам) тупо дропаются, замещаясь результатом интерполяции двух соседних строк четного полукадра. Качество похабное, потеря четкости — драматическая. На ровных наклонных линиях появляются омерзительные "зубцы".

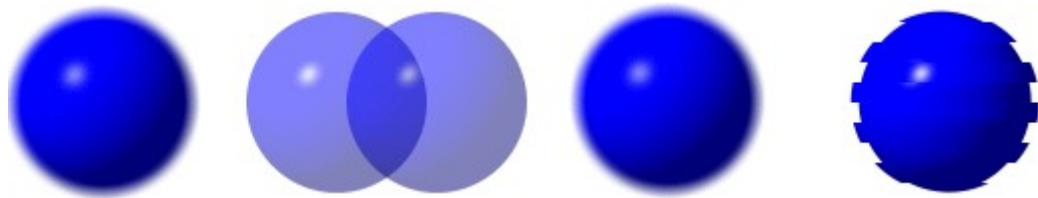


Рисунок 6 слева направо действие фильтров при high motion: "-vf field=0", "linear blend", "median deinterlacing" и аддитивный фильтр с компенсацией движения

Фильтр "linear blend" — линейное смешивание — ("-vf pp=lb"), ничего не дропает, а растягивает четные и нечетные полукадры до полного кадра путем интерполяции, после чего накладывает их друг на друга. Как следствие — на неподвижных или малоподвижных сценах (low motion) мы практически не теряем разрешения (правда, слегка "мылим" картинку за счет интерполяции), но вот на middle motion "мыло" прет со страшной силой и движущиеся объекты утрачивают четкость, а на high motion начинают появляться "призраки" (см. рис. 6).

Медианный фильтр — median deinterlacing filter — ("-vf pp=md") очень похож на линейный и кубический, но вместо того, чтобы выкидывать нечетные строки, он замещает их

усредненным значением вертикальных пикселей двух четных и одной нечетной строки. То есть, если linear blend работает с целыми полукадрами, то median – с отдельными линиями, что быстрее и не гробит весь кадр. Но вот острые объекты и тонкие вертикальные линии корежатся просто ужасно.

"FFmpeg deinterlacer" ("-vf pp=fd") представляет собой своеобразный гибрид blend'a и линейного интерполятора, с той лишь разницей, что он работает только с четными полями, оставляя нечетные нетронутыми, в результате чего наследует лучшие и худшие черты обоих одновременно. Разрешение теряется (пусть и не так сильно), а у быстродвижущихся объектов появляются "призраки" (хоть и не такие заметные как у чистого blend'a).

Адаптивный фильтр Donald'a Graft'a – adaptive kernel deinterlacer filter – ("-vf kerndeint") самый продвинутый. При правильной настройке сцены с low motion останутся практически неискаженными (нет потери разрешения, резкости и фаз движения), но вот middle и high motion являются призраками, притупляют острые углы и все это ценою весьма солидных процессорных ресурсов. Зато мы можем задавать порог изменения пикселей в соседних полукадрах, при котором начнет работать deinterlacer. В случае с упомянутым концентром PAIN'a достаточно разделить разноцветные вспышки прожекторов в соседних полукадрах, а с остальным можно и смириться.



**Рисунок 7 результат действия адаптивного фильтра — исправленное изображение  
(сравните его с [рис. 2](#))**

Порог задается параметром "threshold", принимающим значения от 0 до 255 (по умолчанию 10), причем чем меньше значение, тем агрессивнее себя ведет фильтр. лично мышь предпочитает ставить порог в 27, но это дело вкуса, к тому же сильно зависящее от конкретного видеоматериала. Увидеть пиксели, над которыми поработал deinterlacer можно установив параметр tap в единицу — ("-vf kerndeint=27:1"). Два следующих параметра, будучи установленными в единицу, повышают резкость, убирая мыло, но... общее качество от этого обычно только страдает. Однако, все зависит от конкретного видеоматериала, так что тут надо экспериментировать.

Так какой же фильтр следует применять? Ответ неоднозначен. При слабом ЦП и фильме/клипе снятом в high motion ключе — лучше "-vf field=0", пожалуй, и не придумать. Никаких призраков и минимум мыла, а если еще и карта поддерживает аппаратное сглаживание при масштабировании...

Фильмы с большим кол-вом неподвижных сцен и слабым ЦП – median или FFmpeg. При мощном ЦП — адаптивный фильтр рулит на все 100%, однако, следует помнить, что при

high motion для получения хорошего качества threshold приходится выкручивать за 100, при этом мы получаем ту же самую картинку, что и "-vf field=0".

### **>>> врезка два в одном**

Некоторые фильмы содержат как ч/с, так и прогрессивные сцены и ничего удивительного в этом нет. Допустим, живые актеры снимались на ч/с камеру, а спецэффекты монтировались на компьютере в прогрессивном режиме.

### **>>> врезка преимущества ручной коробки над автоматической**

Mplayer позволяет не только задавать порог срабатывания адаптивного фильтра, но и включать его клавишей <D> (внимание! работает только с видео драйверам xvmc).

### **>>> врезка что, где когда?!**

В прогрессиве идут все кинофильмы (т.е. снятые на пленку) и практически все современные фильмы/клипы. Ч/с режим в основном встречается на записях концертов, спортивных соревнований и т. д.

### **прямой telecine и обратный**

Возьмем DVD-диск, изначально записанный в PAL (25 кадров в сек.), и попытаемся подготовить его для стран, где рулит NTSC (30 кадров в сек.). Вопрос: как быть? Что делать?! Вообще-то этот вопрос возник не вчера и даже не позавчера, а очень давно возник, еще когда фильмы, снятые на пленку (24 кадра в сек.) начали транслировать по PLA/SECAM'у с их 25 кадрами.

А чего тут мудрить и лукавить? 24/25 — слишком малая величина, чтобы ускорение фильма стало заметным. Ну будет 2x часов фильм идти 115 минут вместо положенных 120, ну и что? Больше рекламы поместиться ;) Стоп! А звук... Неслабый несинхрон в 5 минут к концу фильма набегает. То есть Шварц-негр нажимает на курок, а звук выстрела раздается только через... 5 минут! И чтобы зрители не охренели частоту звуковой дорожки увеличивают на  $24/25 = 0,96$ . Уроды! Хрен с ним со Шварцем. А если это оперетта?! Для музыкального уха разница в 0,96 вполне заметна и впечатление уже не то.

Mplayer позволяет решить эту проблему форсированием fps в 24 кадра в секунду ("-fps 24") и ресэмплинга аудиопотока, хотя ресэмплинг в реальном времени меломанам лучше не применять, их уши такого издевательства просто не выдержат. Качество только понизится (и тут без хорошего аудио-редактора не обойтись).

Но вот перевести 24 кадров с пленки в 30 кадров системы NTSC лобовым путем уже не получается. Разница оказывается слишком заметной. Вот и приходится прибегать к отвратительной вещи, именуемой телесинтом (telecine) за описание которой мышь опять-таки отсылает читателей к Википедии — <http://en.wikipedia.org/wiki/Telecine>, раздел "Frame rate differences".

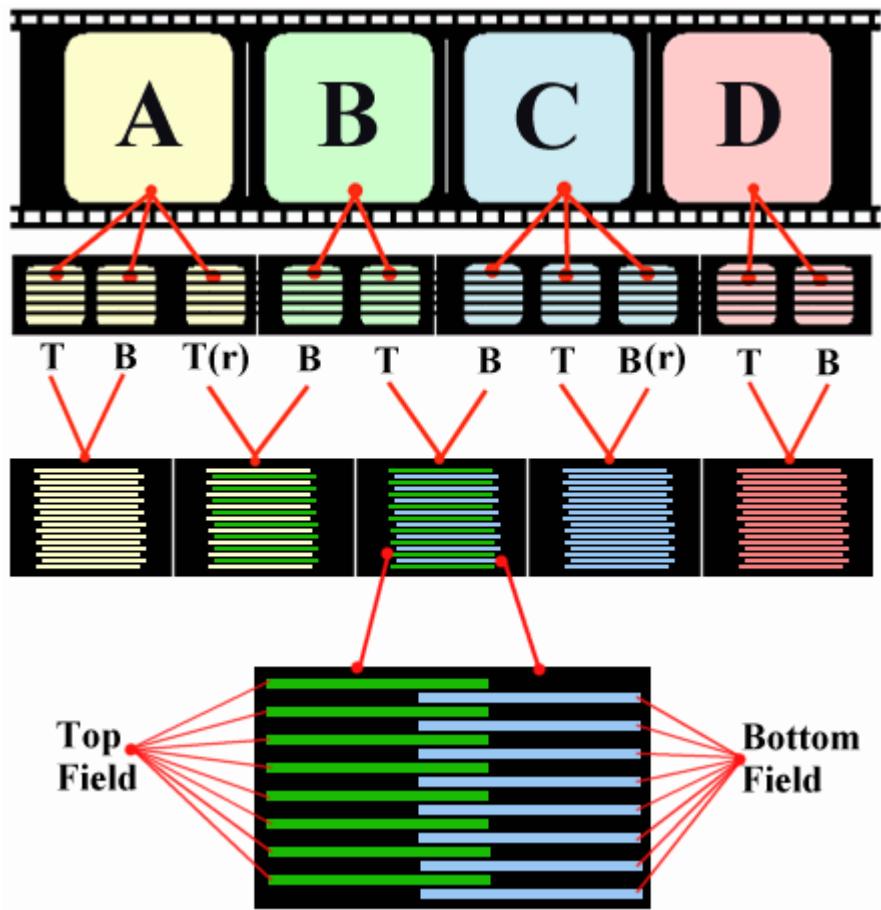


Рисунок 8 смешивание полукадров по классической схеме 2:3 pulldown'е

По классической методике (а есть и другие) 24 кадра разбиваются на 48 полукадров (а в NTSC этих полукадров 60) и каждый второй исходный полукадр дублируется, после чего дублируется каждый третий, затем опять каждый второй и т. д. Отсюда мы получаем схему 2:3:2:3:2:3... или просто "2:3 pulldown". В eng "pull" – тянуть, "down" – вниз. То есть компенсировать увеличение частоты дублированием кадров.

Если взять в руки калькулятор и рассчитать, мы получим, что реальная частота фильма после преобразования составит 23,976 кадра в секунду вместо положенных 24. То есть фильм чуть-чуть замедлится. Совсем немного. На кончик мышного хвоста или даже еще меньше. А вот плавность движений (за счет дублирования кадров) пострадает весьма радикально. Например, когда в фильме "Чужие" (Aliens) показан медленно летящий звездолет, то на оригинальных дисках с ним все ОК, а вот после преобразования в NTSC – создается впечатление, что не хватает мощности ЦП, т. к. звездолет движется рывками.

И вот тут начинается самое интересное. На правильно изготовленном DVD (независимо от того PAL он или NTSC) дублирующиеся кадры быть не должно. По стандарту. И pulldown обязан осуществлять сам DVD-плеер. При необходимости (это так называемый мягкий telecine). А поскольку, телевизоры, работающие только в системе NTSC давно канули в лету, никакой pulldown никому на хрен не нужен и DVD отображается в PAL-режиме, даже если на коробке написано NTSC.

Но вот некоторые дубломы (иначе их не назовешь) выполняют pulldown до записи диска и дублирование кадры физически попадают в видеопоток (жесткий telecine) от чего его размер возрастает, но размер — это не страшно. При этом теряется качество. Тоже самое происходит, если видеофайл записывается с NTSC-канала, по которому передают фильм/клип изначально снятый на пленку или PAL-камеру. И еще ухитряются называть это лицензионными дисками!!! Уроды!

Естественно, на каждый pulldown найдется свой pullup и куча detelecine фильтров, описание которых содержится в справке к mplayer'у и перечислять их здесь нет ни места, ни возможности. Увы, операция обратная telecine'у в общем случае невыполнима, поскольку дублированные кадры никак и ничем не помечены и кроме схемы 2:3 есть еще много других

схем pulldown'a. Автоматика обычно лажает и нужные фильтры приходится находить путем научного перебора. А если видеоматериал еще и черезстрочечный, то это вообще кранты и полный дизастер, подробный разбор которого требует отдельной статьи или даже целой книги.

Тем, кто еще не вкурил в особенности стандартов телевещания (а курить надо именно их), и мышь рекомендует фильтр pullup с параметрами по умолчанию. Этот фильтр, сравнивая соседние кадры, ищет схожести и различия, удаляя ненужные дубликаты, а так же удаляет high-motion кадры испорченные interlacing'ом, что обеспечивает вполне приемлемое качество.

*Внимание! Pullup-фильтр корректно работает только в паре с фильтром softskip, который должен быть указан \_за\_ ним, а частота понижена в 4/5 от оригинальной ("-fps 24000/1001 -vf pullup softskip").*



Рисунок 9 видео, насилиственно подвергнутое жесткому pulldown'y (слева) и результат работы фильтра pullup (справа)

### >>> врезка *россыпи трюков*

Еще одна причина по которой может дергаться изображение и "рваться" звук (или же наблюдаться нарастающий несинхрон изображения и звука, "обнуляющийся" при каждом позиционировании, т. е. перемотке назад/вперед) — это несовпадение частоты, прописанной в заголовке файла, с фактической частотой. mplayer следит за синхронизацией аудио с видео и при необходимости либо дропает либо дублирует кадры.

Задать требуемую частоту можно с помощью уже упомянутого ключа `-fps`, принимающего следующий ряд стандартный значений: 24, 25, 30, 30000/1001, 24000/1001 (вот-вот, именно так, через дробь). Но помимо стандартных значений еще попадаются файлы, записанные на каких-то совершенно диких частотах (причем не тех, что указаны в заголовках). Определить частоту можно как методом перебора, так и аналитически. Mplayer показывает в графе A-V рассинхронизацию между звуком и видео, а так же количество дропнутых кадров для приведения ее в согласование, когда рассинхрон достигнет некоторой критической отметки (см. предпоследнюю цифру справа в строке статуса mplayer'a). Вот мы и калькулируем, если за X секунд мы отстали на Y кадров, то текущую fps нужно умножить на Y/X.

Для любителей просмотра фильма с субтитрами. Часто бывает так, что фильм идет с частотой 25 (классический PAL), а субтитры скаченные с сети, — 23,976 (soft 2:3 pulldown) или 29,97 (NTSC). Конечно, в любом редакторе субтитров их легко конвертнуть, но это будет уже off-line, что не интересно. Mplayer предлагает два решения. Менять частоту самого фильма, согласуя ее с частотой субтитров (ключ `-fps`, заботу по синхронизации звука с видео mplayer возьмет на себя), либо изменить частоту самих субтитров ключом `-subfps`. Первое решение обычно приводит к деградации качества, второе иногда глючит (баг в mplayer'e?) так что на практике приходится использовать оба.

## заключение

Мы рассмотрели лишь две основных проблемы с которыми сталкиваются любители домашнего видео: ч/с режим и жесткий pulldown. А всего их... и по мере роста коллекции видеофайлов проблемы лавинообразно нарастают.

Когда коллекция мышьяка насчитывала сотню дисков он использовал любой плеер, что оказывался под рукой. Но вот число дисков приблизилось к тысяче... К тому времени мышь освоил кучу видео-редакторов и написал множество утилит, фиксящих популярные баги в файлах. Но все это требовало времени на перекодирование. А перекодирование — это офф-лайн.

Сейчас у меня десятки тысяч дисков и еще больше файлов и времени на их перекодирование для просмотра в любом плеере — нет и не будет. Так что mplayer, позволяющий накладывать нужные фильтры налету, превратился в безальтернативный вариант.

### >>> врезка полезные ссылки

- A&E's Technical Guides to All Things Audio and Video:
  - цифровое видео для начинающих (на английском языке):  
<http://www.animenmusicvideos.org/guides/avtech/index.html>;
- DVD Benchmark Part 5 - Progressive Scan DVD:
  - доступно о прогрессиве (на английском языке):  
[http://hometheaterhifi.com/volume\\_7\\_4/dvd-benchmark-part-5-progressive-10-2000.html](http://hometheaterhifi.com/volume_7_4/dvd-benchmark-part-5-progressive-10-2000.html);