

Алгоритм управления битовой скоростью TM5

5 января 2023 г. 1:31

(Алгоритм о том, как можно контролировать битрейт, т.е. каким образом можно получить нужный битрейт без многократного кодирования видео.)

Пусть кодер должен генерировать битрейт видеопотока с

Входные кадры делятся на группы кадров (GOP, groups of frames). Соответственно, мы заранее знаем конфигурацию, т.е. сколько у нас интра-кадров (которые кодируются без остальных) внутри, сколько В-кадров (с двунаправленным предсказанием) и сколько Р-кадров (predicted).

Мы можем вычислить битовые затраты на одну такую группу кадров

$R(I)$, $R(P)$, $R(B)$ - целевые значения числа бит на I, P, B кадры соответственно

$N(P)$ и $N(B)$ - кол-во кадров в группе, мы это заранее знаем

И подразумевается, что например у нас один интра-кадр в группе

Поэтому $R(I) + N(P) \cdot R(P) + N(B) \cdot R(B) = R(GOP)$ - число бит на группу кадров

GOP level RC. $\hat{R}_I + N_P \cdot \hat{R}_P + N_B \cdot \hat{R}_B = R_{GOP}$, где

$R_{GOP} = \frac{1+N_P+N_B}{fps} \cdot C$ - битовый бюджет на каждый GOP, $\hat{R}_I, \hat{R}_P, \hat{R}_B$

- целевые значения числа бит на I, P и B кадры, соответственно,

и $\hat{R}_I \geq \hat{R}_P \geq \hat{R}_B$.

(fps - число кадров в одной секунде)

Также для таких алгоритмов кодирования надо учитывать, что Р-кадры ссылаются на интра-кадры, получается, что чем лучше сжат Р-кадр, тем больше шансов найти хороший референс для каждого блока, т.е. если Р-кадр очень сильно сжат, т.е. он уже с кучей артефактов, искажений, то разностный блок будет плохой, так что желательно сжимать В-кадр лучше, чем Р-кадр. Кроме этого, I-кадр сам по себе сжимается хуже, т.к. не использует другие кадры, что приводит к большим тратам бит на него. В целом мы должны тратить на I-кадры больше бит, чем на Р-кадры, на Р-кадры - больше, чем на В-кадры.

Когда переходим к одному кадру, мы делаем следующее.

Пусть R с домиком - кол-во бит, которые мы хотим потратить на кадр. Как мы можем это сделать?

Мы можем разбить кадр на макроблоки, т.е. на минимальные единицы, к которым мы можем применить шаг квантования. Заводим так

называемый виртуальный буфер, число бит в котором вычисляется так

Frame level RC. \hat{R} - обеспечивается при помощи виртуального

буфера, т.е., для каждого макроблока вычисляется число бит в

буфере b_i как

$$b_{i+1} = b_i + r(q_{i+1}) - \frac{\hat{R}}{M}$$

где M - число макроблоков в кадре, и шаг квантования q_{i+1}

вычисляются как

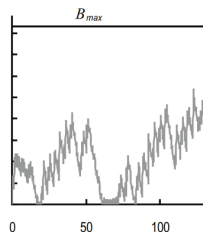
$$q_{i+1} = \alpha \cdot \frac{b_i}{B_{max}}$$

где α - константа, и B_{max} - максимальное число бит в буфере.

Смысл формул:

Постоянно добавляем кол-во бит в виртуальный буфер и вычитаем в зависимости от этого

Чем больше бит в буфере, тем дробь ближе и ближе к единице (т.е. тем сильнее сжимаем). Мы пытаемся кодировать так, чтобы не было переполнения буфера и его опустошения



- Если в процессе кодирования $0 \leq b_i \leq B_{max}$, то

$$\sum_{i=1}^M \frac{\hat{R}}{M} \leq \sum_{i=1}^M r(q_i) \leq \sum_{i=1}^M \frac{\hat{R}}{M} + B_{max},$$

или

$$\hat{R} \leq R \leq \hat{R} + B_{max}.$$

TM5 подходит в случае real time video. Он наполняет буфер и передает его. Чем больше буфер, тем плавнее воспроизведение. Чем меньше буфер, тем меньше задержка, пользователь будет ждать меньше времени. Нужно решать задачу, учитывая это.

Также надо оценивать пропускную способность.

определяется
max задержкой,
которую декодеру
нужно подождать,
прежде чем
начать декодировать

д. кол-во бит в буфере
max размер буфера

