



技术开启新“视”界
Technology Bring New Vision

实时视频通信质量评价及 寻找视频质量甜点的方法

Vik Qian 菊风 (Juphoon) 2018

提纲 & 摘要

- Part1: 实时视频通信质量评价的方法研究
- Part2: 根据质量模型设计质量甜点算法

本演讲将分享 Juphoon 媒体引擎开发团队在视频通信质量上的探索实践，通过对视频通话视频的大量统计分析，采用回归、支持向量机等机器学习方法，分析各种因素对通话质量的影响。

听众的预期收获：

- 通过这个学习交流，你可以了解我们通过统计学方法得到的一些编码相关的质量规律
- 了解我们是如何实现实时的质量客观评价和监控的
- 如果你对视频质量技术一无所知，这个分享试图在较短的时间内让你向在这个领域成为专家更近一步

Part1: 动机



- 有没有一种靠谱的评价视频通话质量自动化方法？
- 能否做到实时监控视频通话的端到端质量？



为什么现有简单的方法不可直接用？

- PSNR、SSIM ---- need Pic by Pic

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right)$$

$$SSIM(x, y) = [l(x, y)^\alpha \cdot c(x, y)^\beta \cdot s(x, y)^\gamma]$$

$$MSE = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2$$

$$c(x, y) = \frac{2\sigma_x \sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}$$

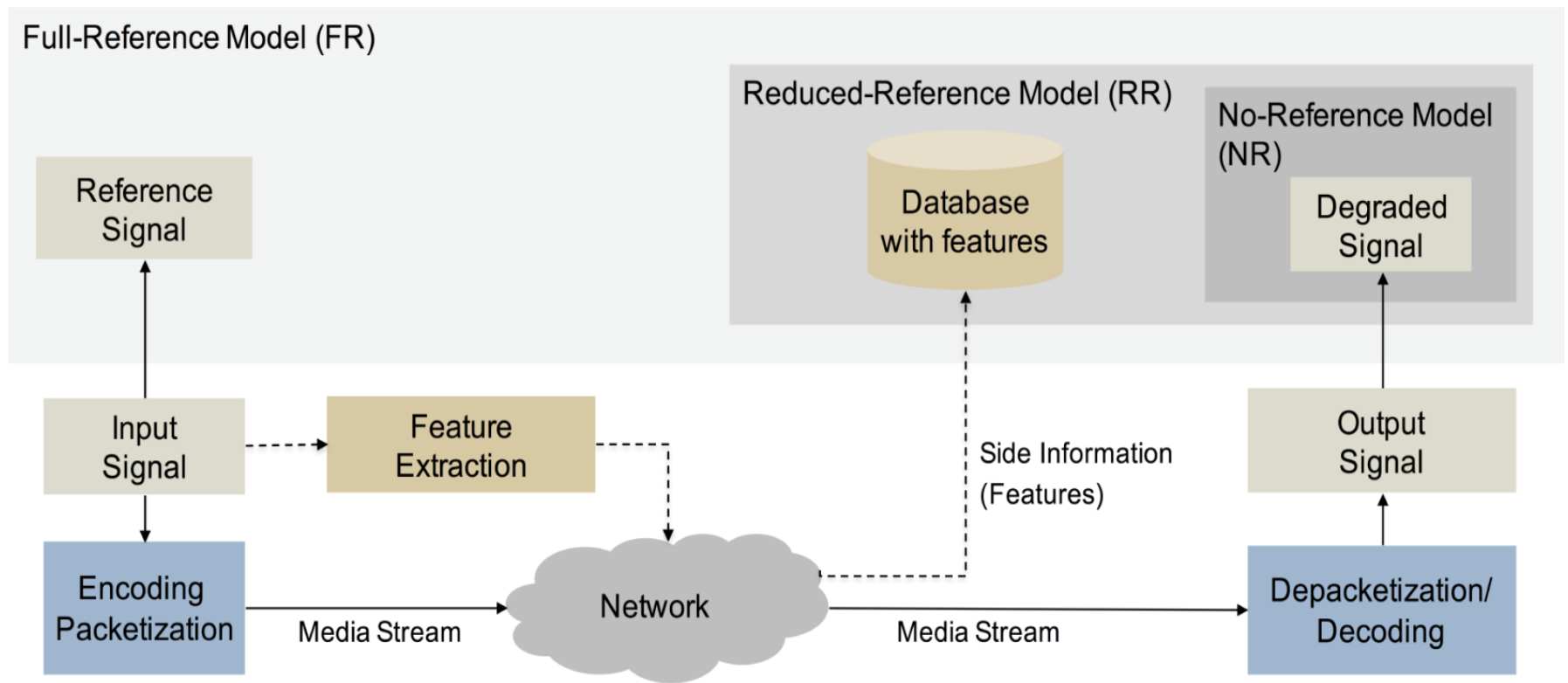
$$s(x, y) = \frac{\sigma_{xy} + c_3}{\sigma_x \sigma_y + c_3}$$

$$l(x, y) = \frac{2\mu_x \mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1}$$

PEVQ、VQuad-HD、VMAF怎么样？

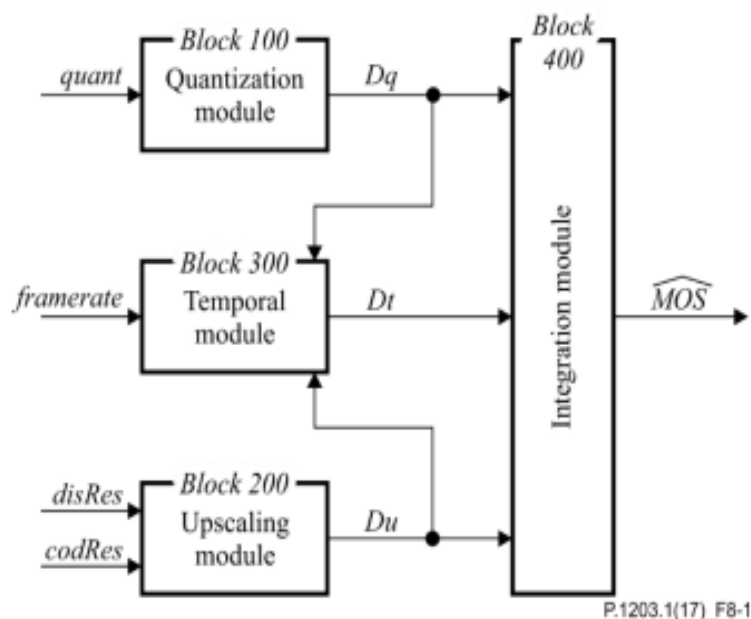
- PEVQ - (ITU-T Rec. J.247 (FR), 2008)
- VQuad-HD - (ITU-T Rec. J.341 (FR), 2011)
- VMAF - (NetFlix, 2017)
- 需要全参考 ---- 不适合实时视频通话

FR\RR\NR -- 3类参考模型



图来自Wikipedia

ITU 基于码流的可靠传输评价模型（2017）怎么样？

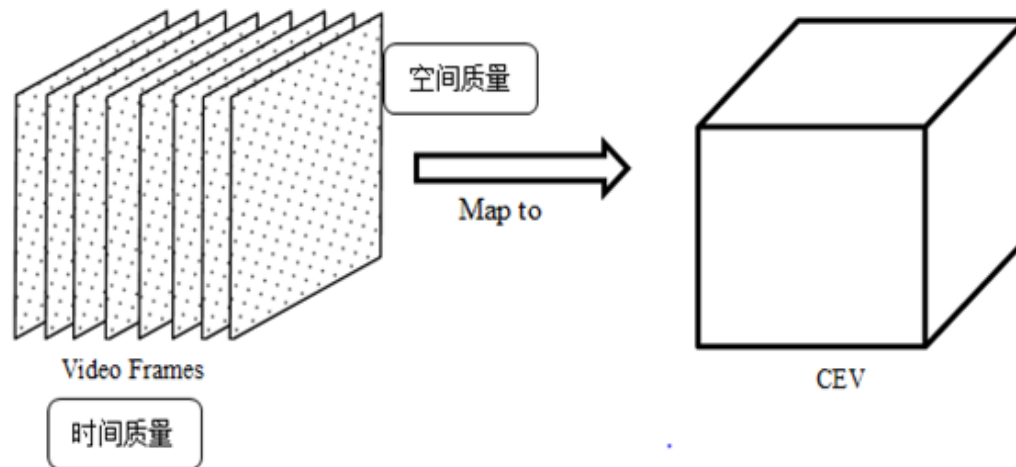


比较接近实时视频通信的质量评价需求，但没有考虑网络丢包、带宽限制、延时。

还不能直接使用。

Pv module in the context of building blocks of the ITU-T P.1203 model

CEV模型(Cubic information Entropy Volume)



$$MOS = TMOS^a * SMOS^b$$

基于 CEV 质量评价模型实践



为什么需要生成系统？

我们研究：

编码失真

量化失真（编码器 QP取值）

频域变换失真（DCT）

下采样失真

超分辨失真（良性）

滤波失真（良性）

传输失真

降低帧率（抽帧）

卡顿（丢帧）

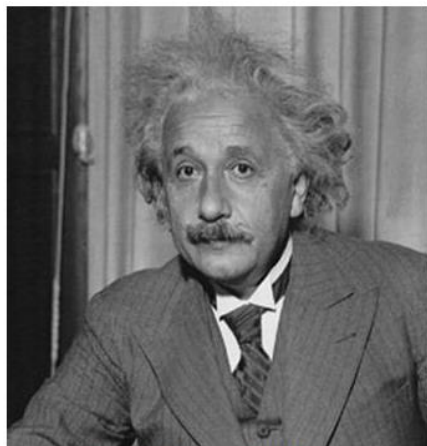
延迟

标准库的各种失真

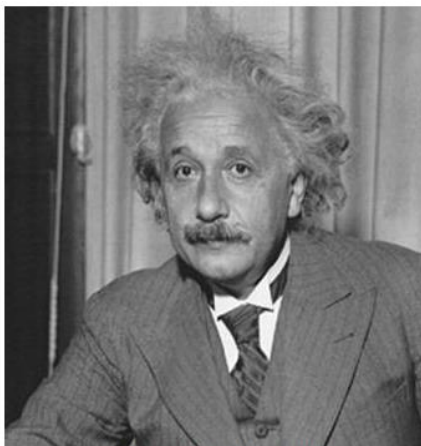
T3*	T8*	C*	L*	Distortion types
✓	✓	✓	✓	Additive Gaussian noise
✓	✓	-	-	Additive noise (more intensive color components)
✓	✓	-	-	Spatially correlated noise
✓	✓	-	-	Masked noise
✓	✓	-	-	High frequency noise
✓	✓	-	-	Impulse noise
✓	✓	-	-	Quantization noise
✓	✓	✓	✓	Gaussian blur
✓	✓	-	-	Image denoising
✓	✓	✓	✓	JPEG compression
✓	✓	✓	✓	JPEG2000 compression
✓	✓	-	-	JPEG transmission errors
✓	✓	-	-	JPEG2000 transmission errors
✓	✓	-	-	Non eccentricity pattern noise
✓	✓	-	-	Local block-wise distortions of different intensity
✓	✓	-	-	Mean shift (intensity shift)
✓	✓	✓	-	Contrast change
✓	-	-	-	Change of color saturation
✓	-	-	-	Multiplicative Gaussian noise
✓	-	-	-	Comfort noise
✓	-	-	-	Lossy compression of noisy images
✓	-	-	-	Image color quantization with dither
✓	-	-	-	Chromatic aberrations
✓	-	-	-	Sparse sampling and reconstruction
-	-	-	✓	Fast Fading Rayleigh

*T3: TID2013, T8: TID2008, C: CSIQ, L: LIVE

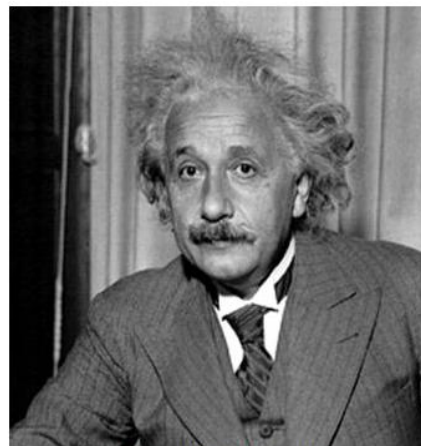
不同失真图片的评价指标



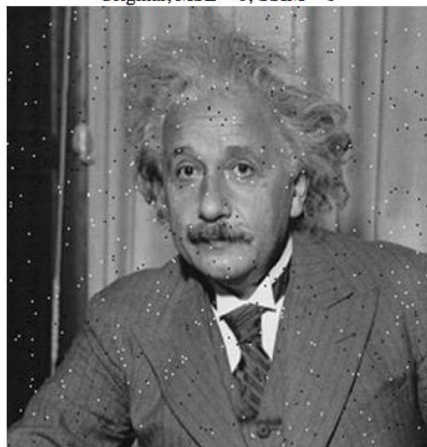
Original, MSE = 0; SSIM = 1



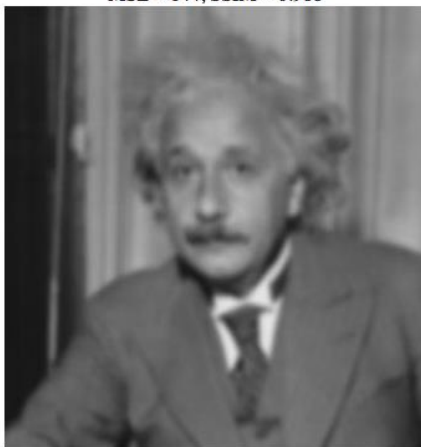
MSE = 144, SSIM = 0.988



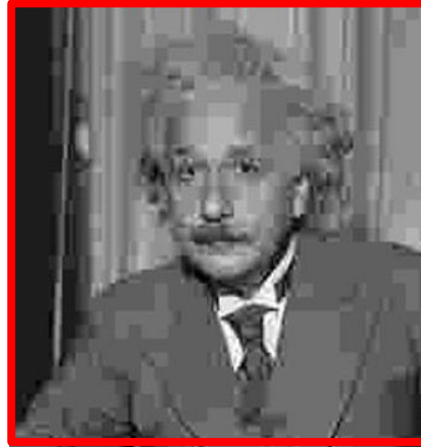
MSE = 144, SSIM = 0.913



MSE = 144, SSIM = 0.840

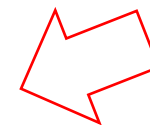


MSE = 144, SSIM = 0.694



MSE = 142, SSIM = 0.662

研究对象



自己生成 VS 公共数据集

自己生成的数据:

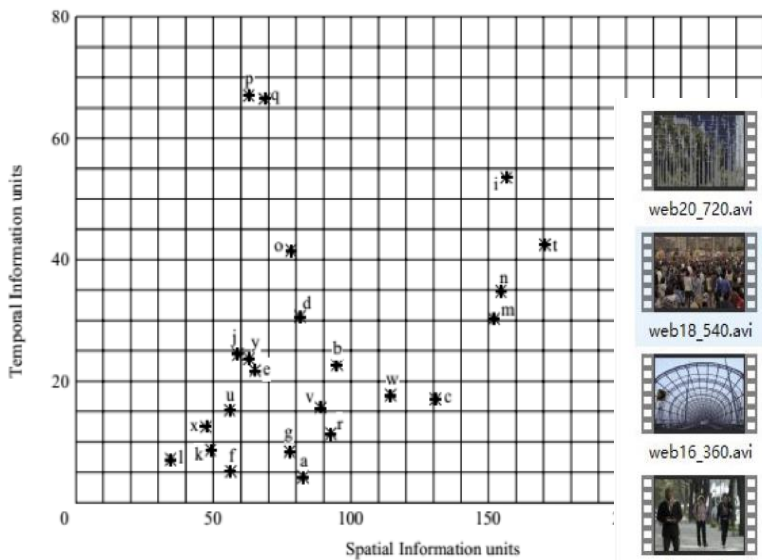
- 针对x264、x265、openh264、VPX视频编码失真做研究
- 传输失真（主要研究时间质量） 600 Videos, 20 Observers
- 编码失真（主要研究空间质量） 300 videos, 15 Observers

Database	Source Images	Distorted Images	Distortion Types	Image Type	Observers
TID2013	25	3000	24	color	971
TID2008	25	1700	17	color	838
CSIQ	30	866	6	color	35
LIVE	29	779	5	color	161
IVC	10	185	4	color	15
Toyama-MICT	14	168	2	color	16
A57	3	54	6	gray	7
WIQ	7	80	5	gray	60

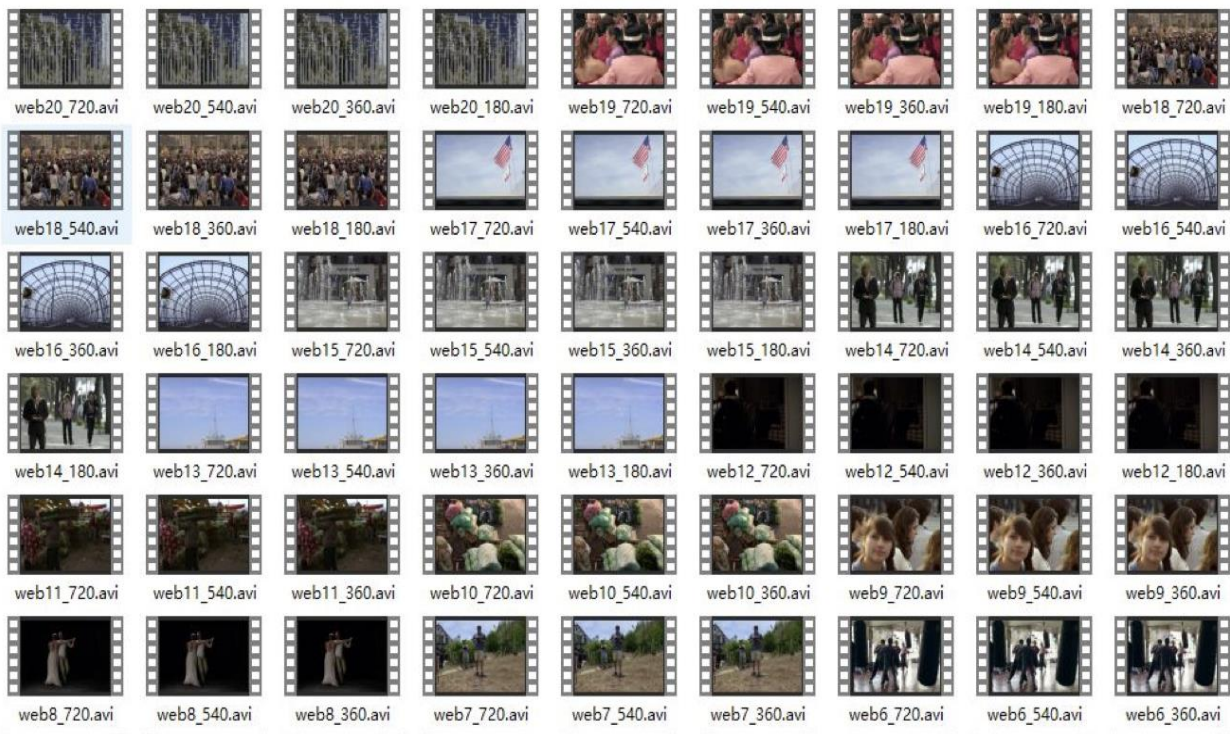
公开的库范围很广，包含各种失真，针对性弱一些。

生成系统

无损视频序列



不同视频源的时空复杂度



众包打分系统

🏠

MOS打分

×

📖

MOS分说明

×

+

用户: 1 [修改用户名](#)

[视频301](#)

[视频311](#)

[视频321](#)

[视频331](#)

[视频341](#)

[视频351](#)

[视频361](#)

[视频371](#)

[视频381](#)

[视频391](#)

[视频401](#)

[视频411](#)

[视频421](#)

[视频431](#)

[视频441](#)

[视频302](#)

[视频312](#)

[视频322](#)

[视频332](#)

[视频342](#)

[视频352](#)

[视频362](#)

[视频372](#)

[视频382](#)

[视频392](#)

[视频402](#)

[视频412](#)

[视频422](#)

[视频432](#)

[视频442](#)

[视频303](#)

[视频313](#)

[视频323](#)

[视频333](#)

[视频343](#)

[视频353](#)

[视频363](#)

[视频373](#)

[视频383](#)

[视频393](#)

[视频403](#)

[视频413](#)

[视频423](#)

[视频433](#)

[视频443](#)

[视频304](#)

[视频314](#)

[视频324](#)

[视频334](#)

[视频344](#)

[视频354](#)

[视频364](#)

[视频374](#)

[视频384](#)

[视频394](#)

[视频404](#)

[视频414](#)

[视频424](#)

[视频434](#)

[视频444](#)

[视频305](#)

[视频315](#)

[视频325](#)

[视频335](#)

[视频345](#)

[视频355](#)

[视频365](#)

[视频375](#)

[视频385](#)

[视频395](#)

[视频405](#)

[视频415](#)

[视频425](#)

[视频435](#)

[视频445](#)

[视频306](#)

[视频316](#)

[视频326](#)

[视频336](#)

[视频346](#)

[视频356](#)

[视频366](#)

[视频376](#)

[视频386](#)

[视频396](#)

[视频406](#)

[视频416](#)

[视频426](#)

[视频436](#)

[视频446](#)

[视频307](#)

[视频317](#)

[视频327](#)

[视频337](#)

[视频347](#)

[视频357](#)

[视频367](#)

[视频377](#)

[视频387](#)

[视频397](#)

[视频407](#)

[视频417](#)

[视频427](#)

[视频437](#)

[视频447](#)

[视频308](#)

[视频318](#)

[视频328](#)

[视频338](#)

[视频348](#)

[视频358](#)

[视频368](#)

[视频378](#)

[视频388](#)

[视频398](#)

[视频408](#)

[视频418](#)

[视频428](#)

[视频438](#)

[视频448](#)

[视频309](#)

[视频319](#)

[视频329](#)

[视频339](#)

[视频349](#)

[视频359](#)

[视频369](#)

[视频379](#)

[视频389](#)

[视频399](#)

[视频409](#)

[视频419](#)

[视频429](#)

[视频439](#)

[视频449](#)

[视频310](#)

[视频320](#)

[视频330](#)

[视频340](#)

[视频350](#)

[视频360](#)

[视频370](#)

[视频380](#)

[视频390](#)

[视频400](#)

[视频410](#)

[视频420](#)

[视频430](#)

[视频440](#)

[第一组>>](#)

[第二组>>](#)

●

[第三组>>](#)

[MOS分说明>>](#)

🔍

播放视频11

×


📖

MOS分说明

×

+

视频73



⏸

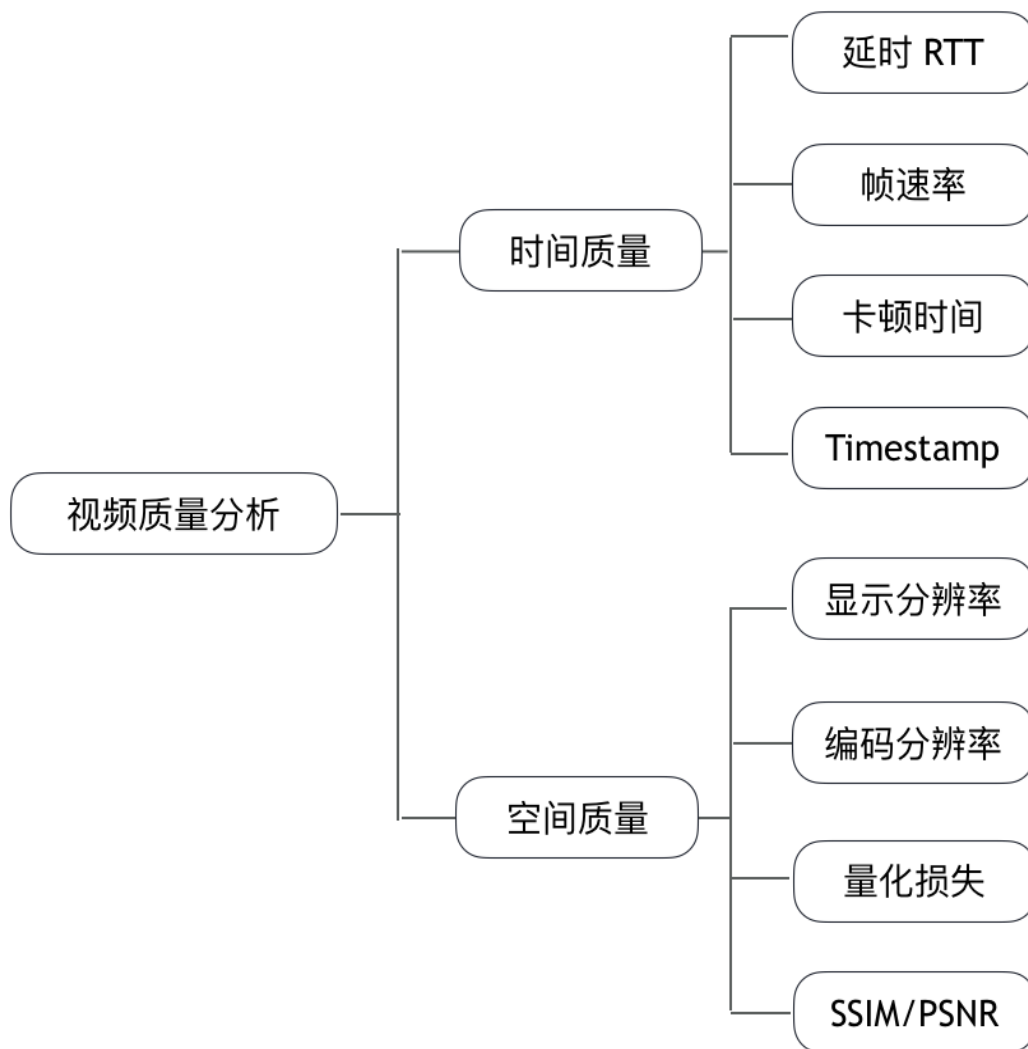
0:03

🔊

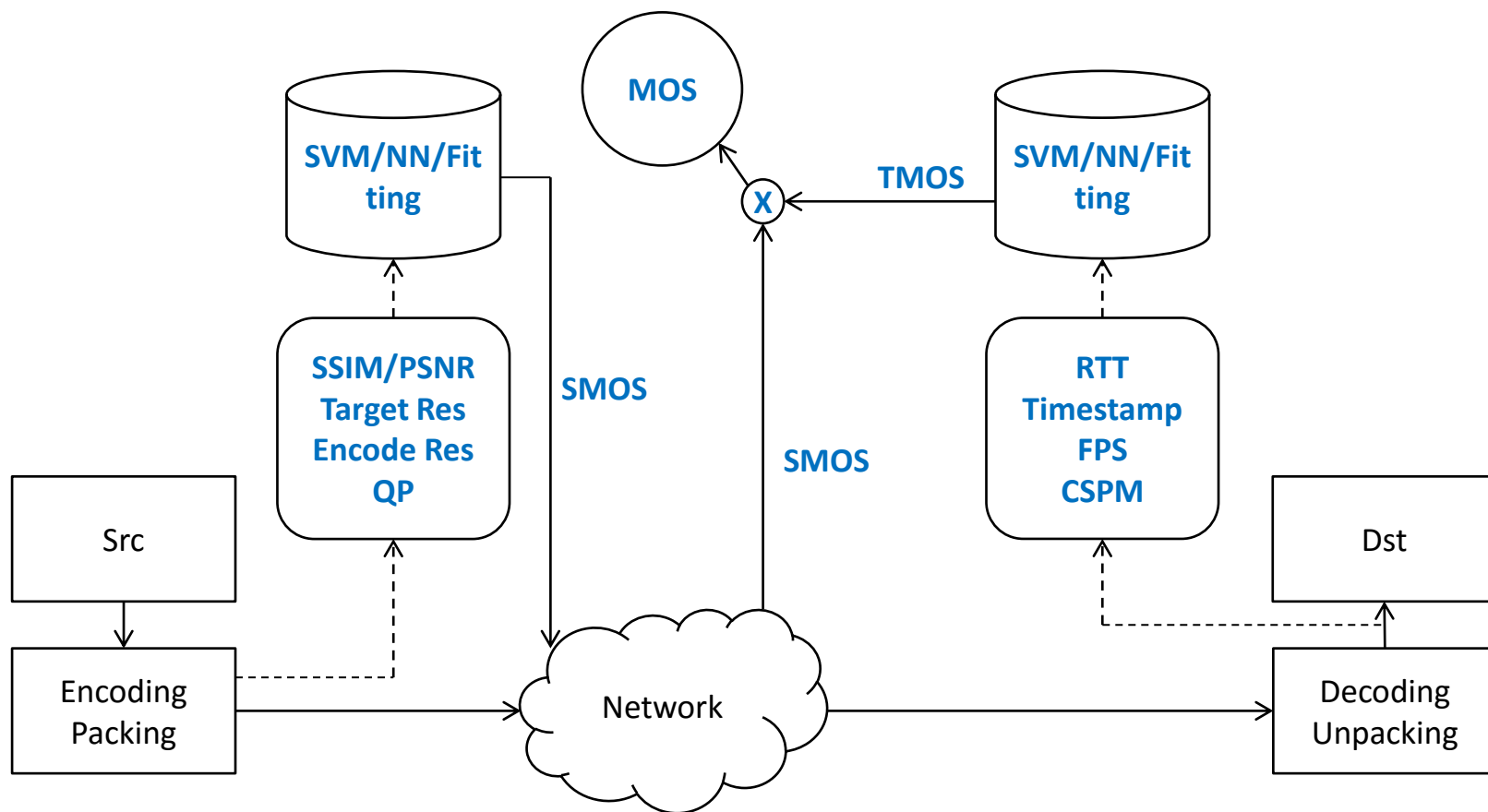
🖼

1打分: 分 [提交](#) 备注: 最小1分, 最大5分, 比如4.5

CEV 影响因素分析



模型实现



分析时间质量TMOS

- RTT
- Play Timestamp
- Frame rate
- CSPM (Congest Seconds Per Minute)

Proposed TMOS Fitting Model

$$TMOS = 5 \times \frac{F_{mos}^{m9} \cdot RTT_{mos}^{m10} \cdot C_{mos}^{m13}}{5^{m9+m10+m13}}, \quad TMOS \in (1,5)$$

$$F_{mos} = m5 \cdot F^2 + m6 \cdot F + m7$$

$$RTT_{mos} = m3 \cdot \ln RTT + m4$$

$$C_{mos} = m11 \cdot CSPM + m12$$

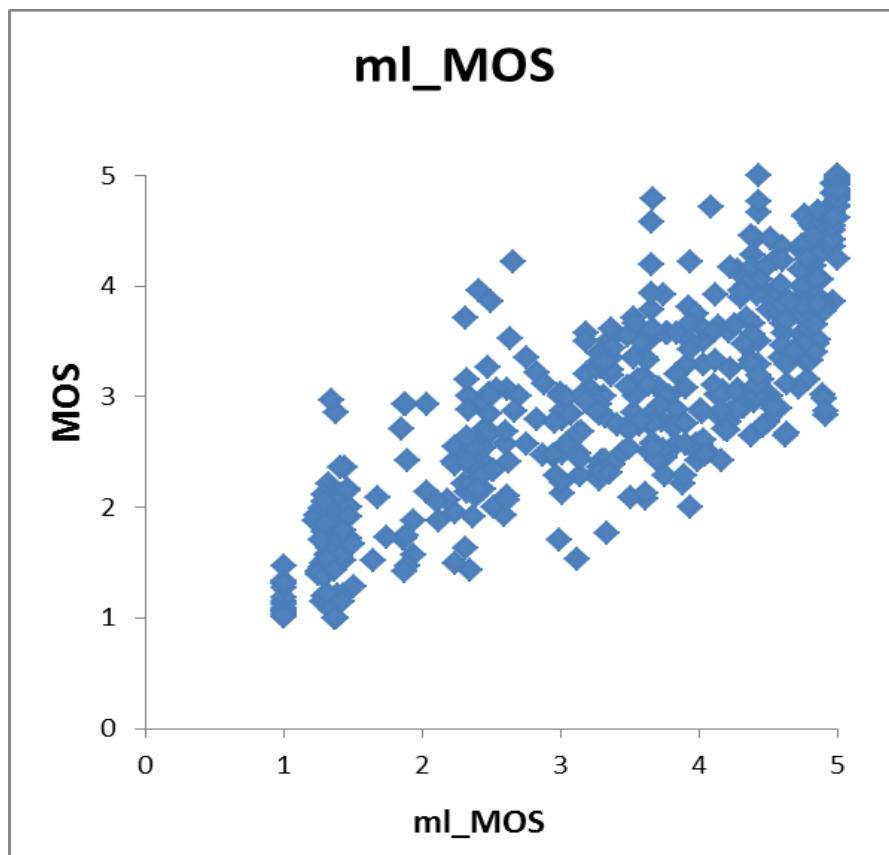
F 为帧率

RTT 为来回延迟

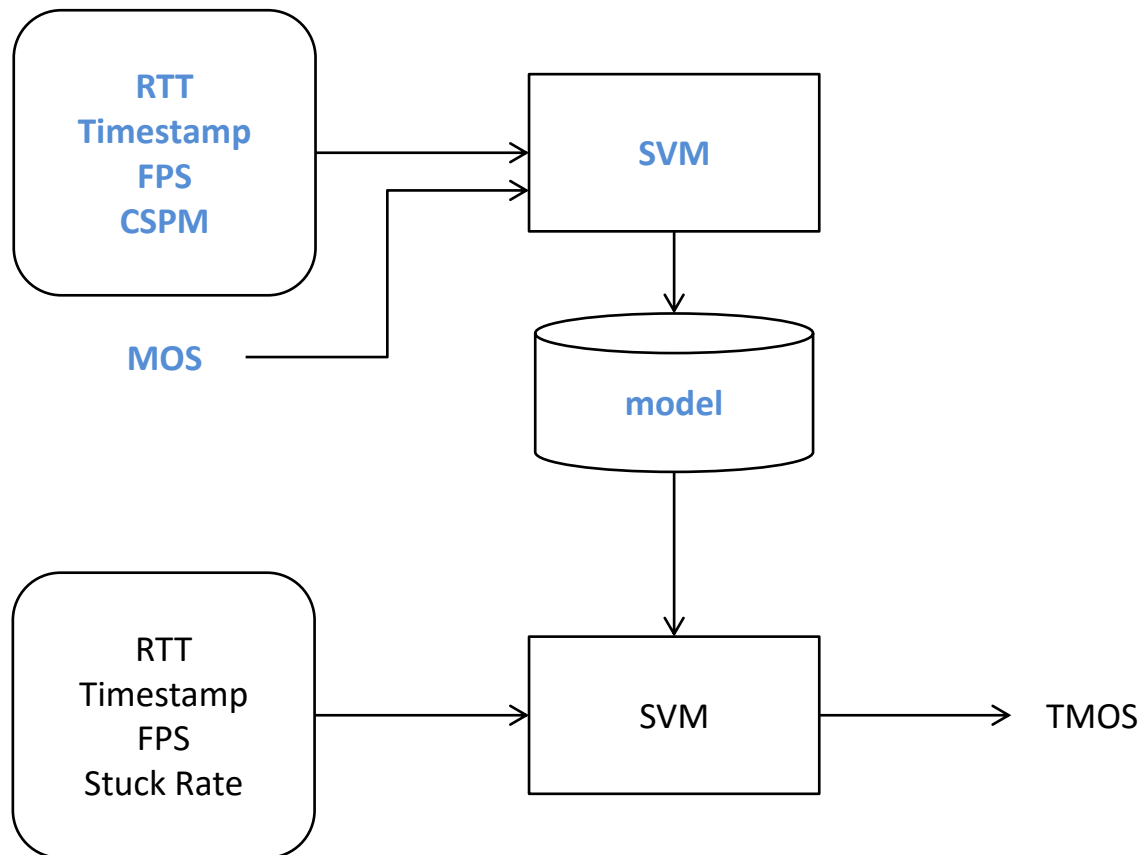
CSPM 为每分钟卡顿秒数

m3=-0.887, m4=8.9061, m5=-0.0048, m6=0.2907, m7=0.6651, m8=0.3, m9=0.5, m10=1,
m11=-0.0667, m12=5, m13=0.5

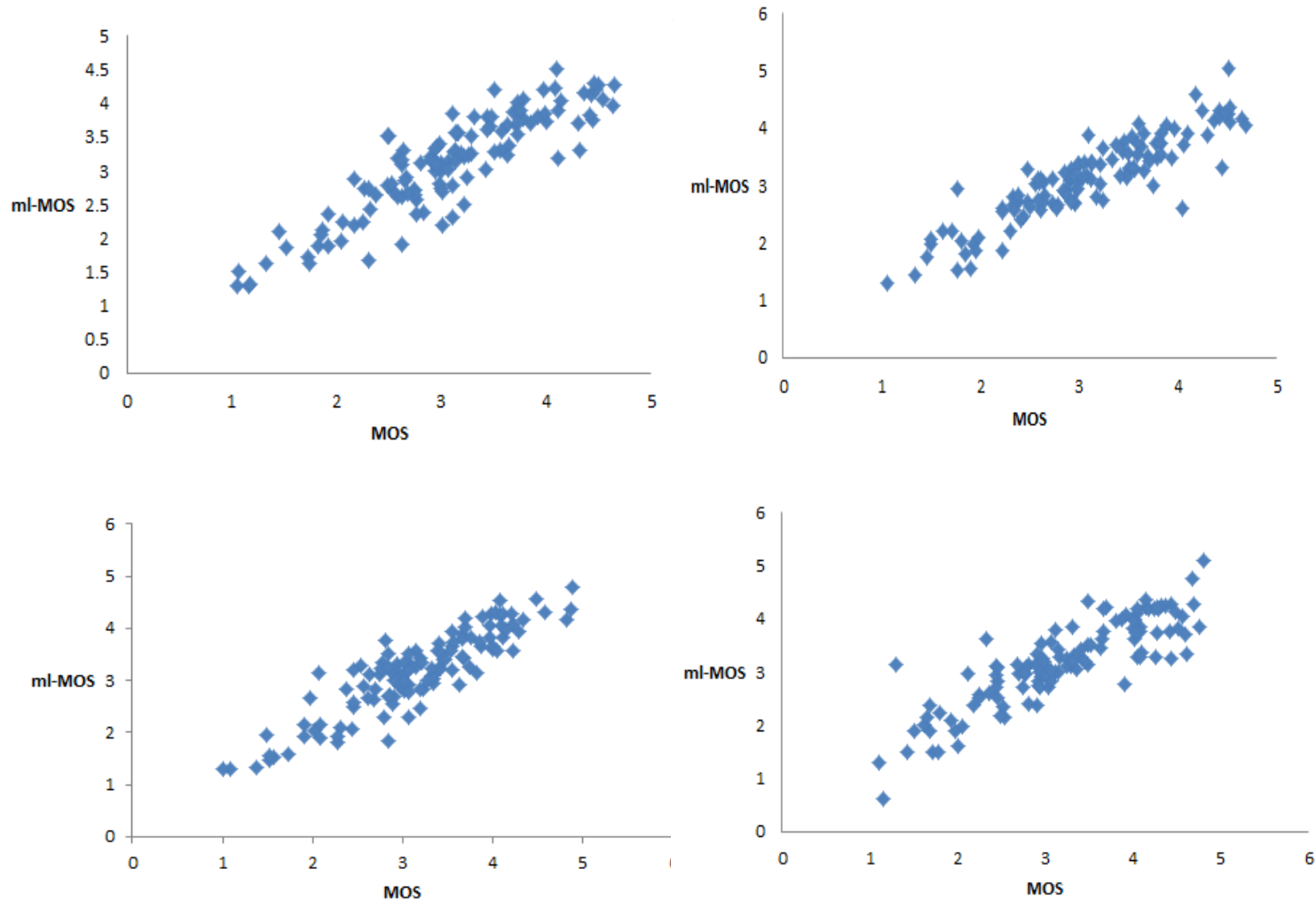
TMOS Fitting 模型效果评估 (Pearson 0.86)



ML – Train by SVM



TMOS SVM效果评估(Pearson 0.93)



TMOS 效果评估

回归预测

- MOS相关系数 0.86
- 所需代码数量极少，速度快
- 需人工设定每个参数占比

可应用于现网质量跟踪

SVM建模

- MOS相关系数 0.93
- 代码量稍多
- 加载时训练模型时间较长

可应用于实验室测试评估

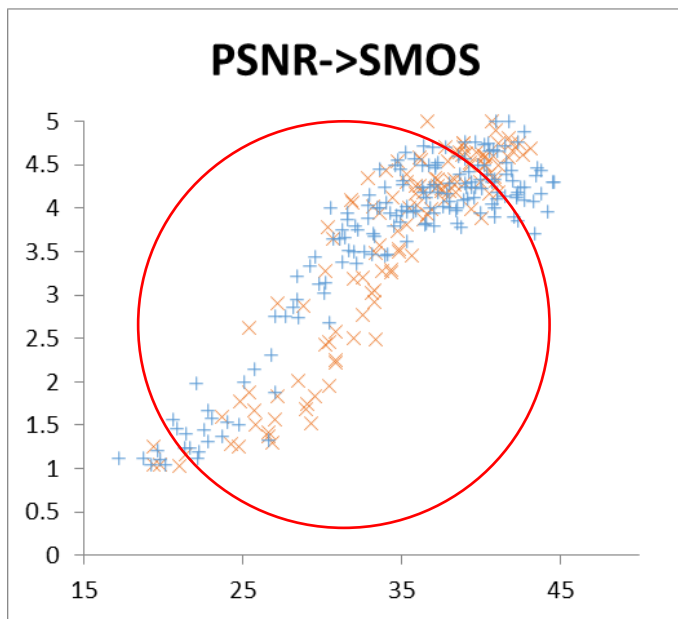
NN 神经网络

- MOS相关系数 0.88 (训练数据不足)
- 需大量数据
- 速度较慢

分析空间质量 SMOS

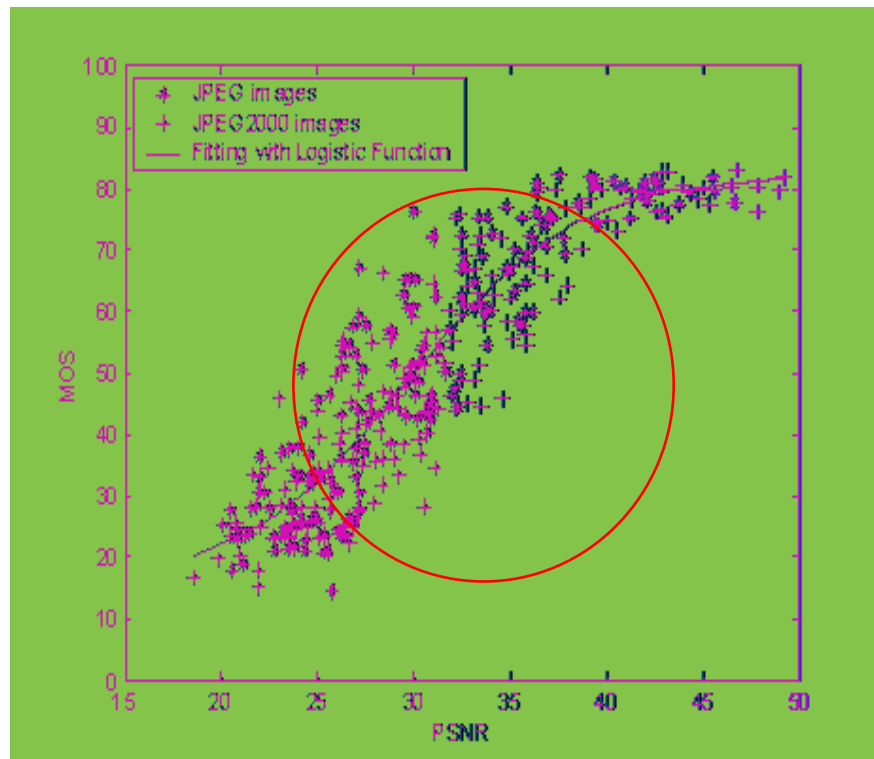
- PSNR/SSIM
- Encode QP
- Target Res
- Encode Res

PSNR to SMOS



Our data

蓝色为360p以下、棕色为360p以上视频



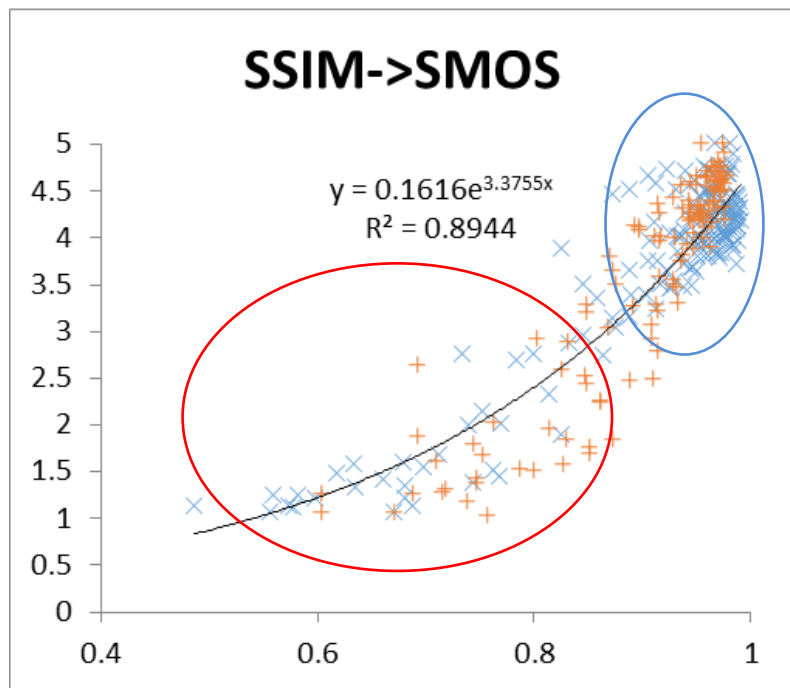
Lab for Image and Video Engineering (LIVE)

University of Texas at Austin

PSNR 两个数据集分布规律比较一致

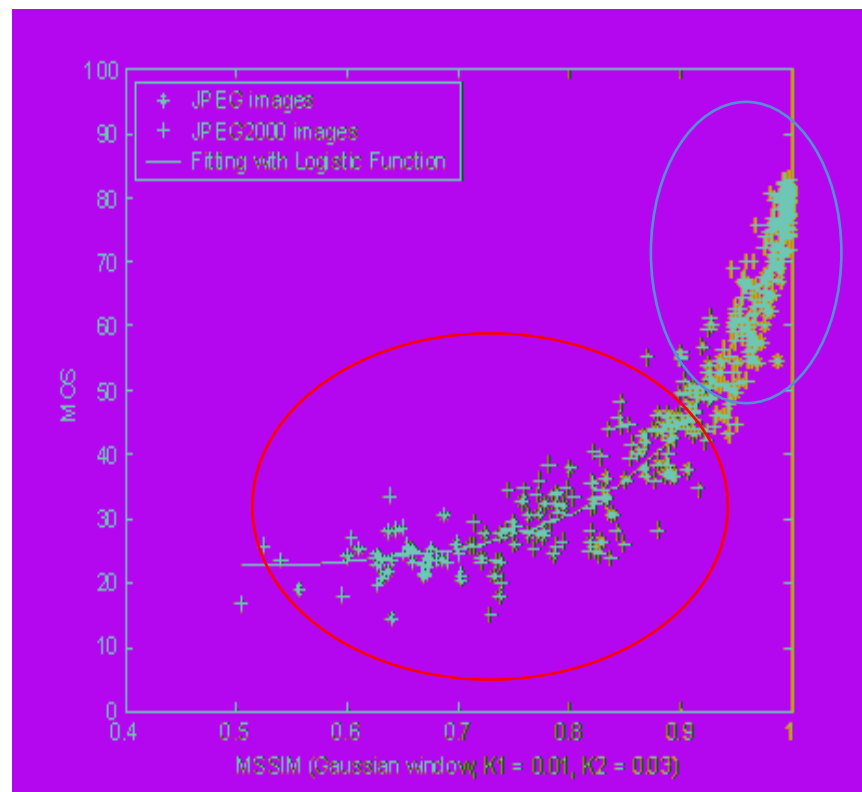
$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}}$$

SSIM to SMOS



Our data

蓝色为360p以下、棕色为360p以上视频



Lab for Image and Video Engineering (LIVE)

University of Texas at Austin

SSIM 比较适合分辨适中及低质量的编码图像，高质量图像分辨能力偏低

SMOS几个实用的数据

SMOS	PSNR	SSIM
4.50	40.6	0.98
4.00	36.7	0.9485
3.00	30	0.8715
2.00	24.5	0.7625
1.50	22.15	0.685

* 不同视频源会有不同，更高分辨率的视频同样
SMOS对应更高的PSNR和SSIM

PSNR VS SSIM

Video	PSNR	SSIM
Average	0.9351	0.9649
All Pearson	0.8960	0.9005
All Fit	0.9104	0.9126
Per Res Fit	0.94569	0.94236

$$S_{mos} = L / (e^{k * (a * psnr + b * rer + c)} + d) + e$$

$$S_{mos} = n5_i \cdot e^{n6_i * SSIM}$$

- 对于同一个视频源不同码率编码的质量评价 SSIM 比较准确 (0.9351 VS 0.9649)
- 全局评价 (有不同视频源) PSNR 和 SSIM 准确度差不多 (0.8960 VS 0.9005)
- 按不同分辨率分类拟合能够提高评价准确度 (0.94569 VS 0.94236)

* All data are PLCC(Pearson)

* rer: Resolution root

Video	PSNR	SSIM
180p-1 x264	0.92957	0.99124
180p-1 x265	0.89507	0.98096
180p-2 x264	0.93219	0.97743
180p-2 x265	0.93099	0.98918
180p-3 x264	0.89999	0.97610
180p-3 x265	0.84779	0.96665
180p-4 x264	0.90696	0.98375
180p-4 x265	0.91696	0.98347
360p-1 x264	0.98373	0.96854
360p-1 x265	0.97711	0.97686
360p-2 x264	0.95902	0.97133
360p-2 x265	0.93746	0.99218
360p-3 x264	0.90630	0.96961
360p-3 x265	0.83201	0.91705
360p-4 x264	0.99041	0.96004
360p-4 x265	0.94688	0.97349
720p-1 x264	0.94722	0.98469
720p-1 x265	0.92224	0.97748
720p-2 x264	0.95470	0.93365
720p-2 x265	0.96963	0.94442
720p-3 x264	0.96404	0.94488
720p-3 x265	0.93483	0.97016
1080p-1 x264	0.94849	0.92598
1080p-1 x265	0.95464	0.93126
1080p-2 x264	0.95660	0.96673
1080p-2 x265	0.90554	0.94717
1080p-3 x264	0.96706	0.93417
1080p-3 x265	0.96580	0.97968

Proposed SMOS Fitting Model

$$SMOS = 5 \times \frac{Smos^{n9} \cdot Remos^{n10}}{5^{n9+n10}}, \quad TMOS \in (1,5)$$

其中 Smos 算法3 档速度（3选一）

Almost No cost

$$PSNR' = n1 \cdot QP + n2 + n3 \cdot rer,$$

$$PSNR' \sim PSNR$$

Fast (*10ms/frame)

$$Smos = L / (e^{k \cdot (a \cdot psnr + b \cdot rer + c)} + d) + e$$

Slow (*20ms/frame)

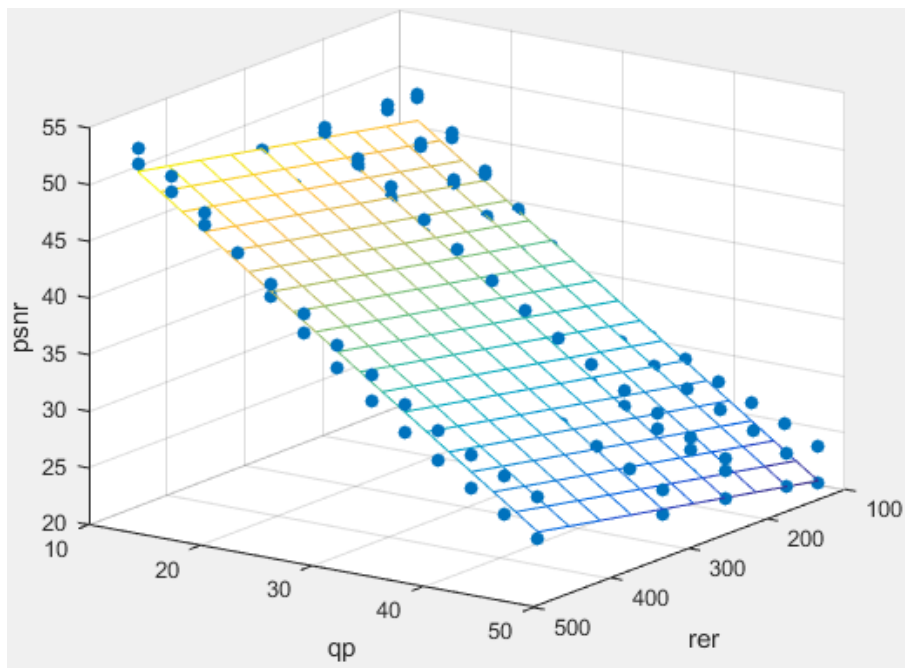
$$Smos = n5_i \cdot e^{n6_i \cdot SSIM}$$

$$Remos = n7 * \ln \left(\frac{TargetRes}{EncodeRes} \right) + n8,$$

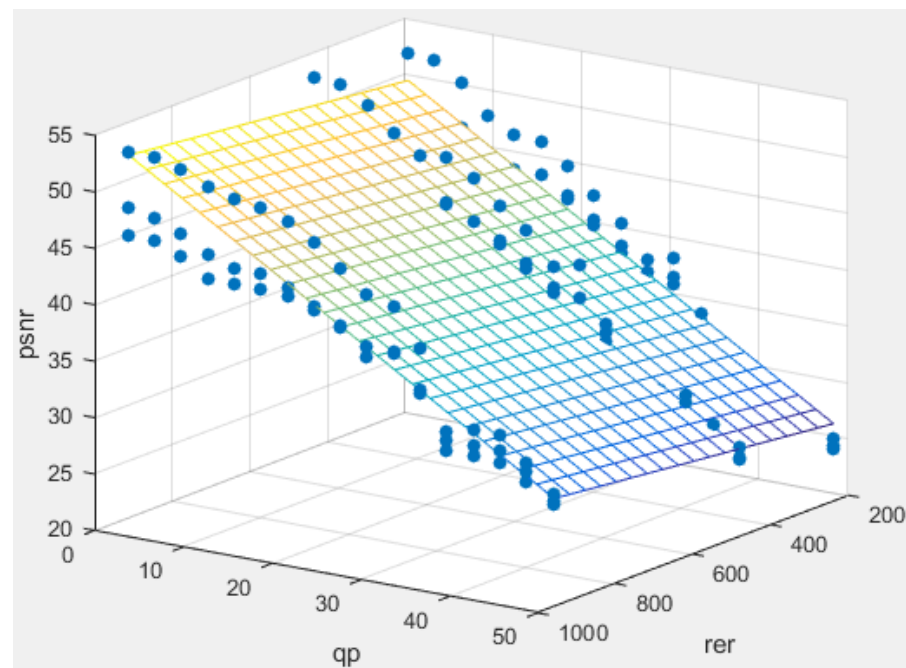
$$Remos \in (1,5)$$

* 数量级参考，不同分辨率、平台速度不同

From QP/rer to PSNR



X264 (PLCC 0.987)

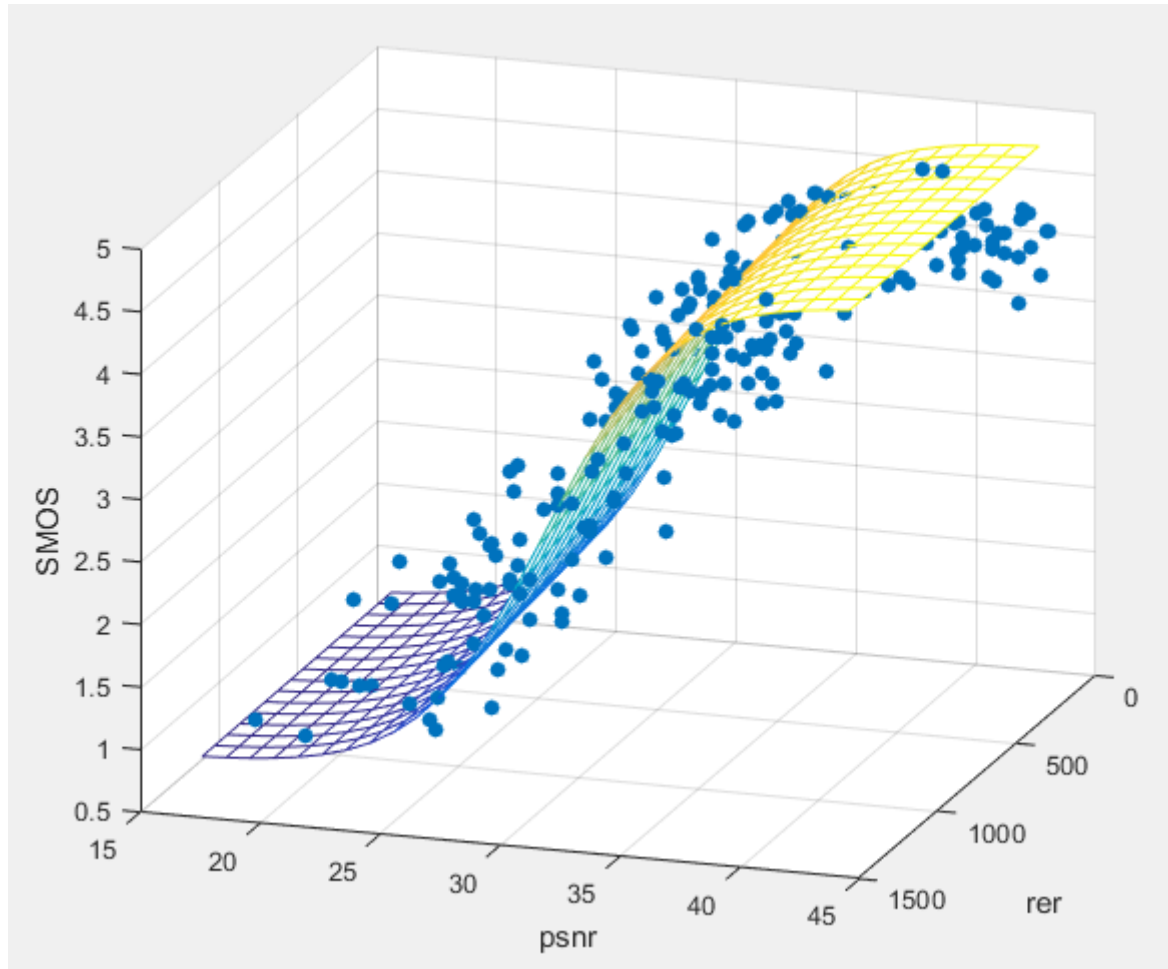


openh264 (PLCC 0.939)

上图为不同视频的H.264编码的 PSNR 和 均值 QP 关系

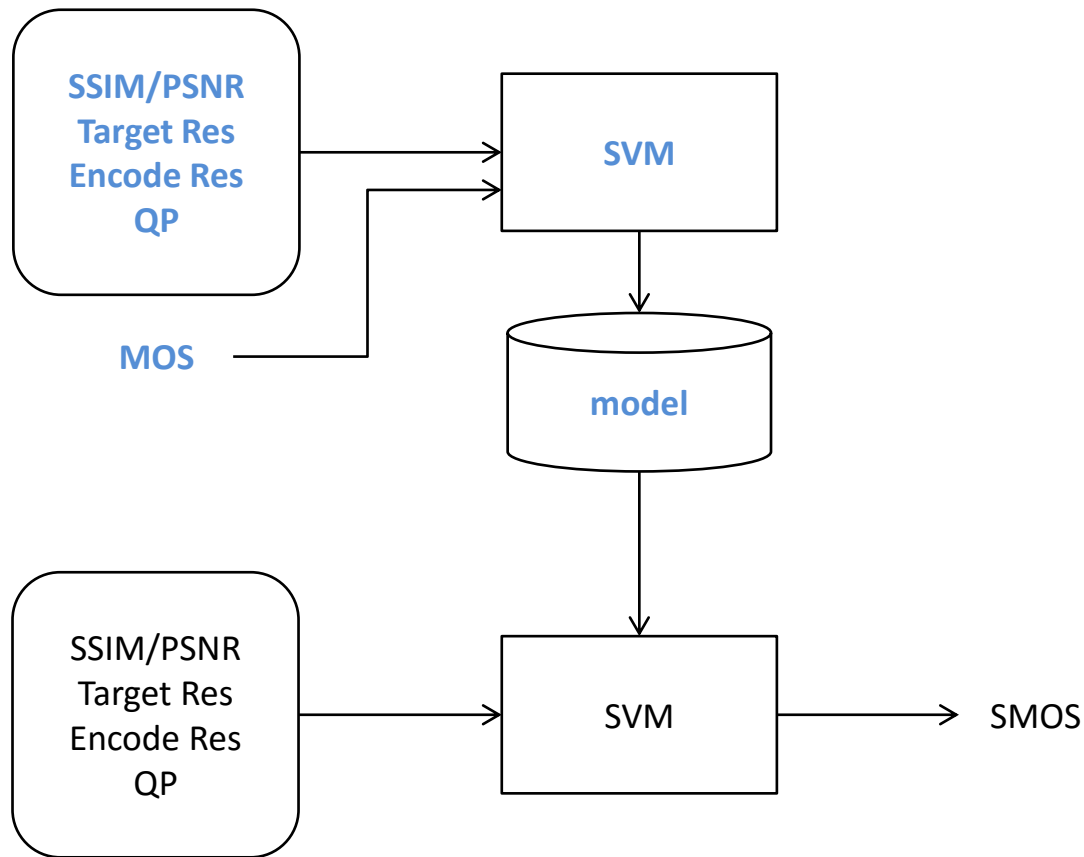
注：不同编码器的QP和PSNR关系会有所不同，需根据实际情况调整

From PSNR/rer to SMOS



PLCC 0.94569

ML – Train by SVM



Proposed CEV MOS

$$MOS = 5 * \frac{TMOS^a * SMOS^b}{5^{a+b}}$$

$$a = 0.6, \quad b = 0.4$$

现网应用：实时通话质量统计

均值趋势 

全部国家▼

全部频道▼

全部平台▼

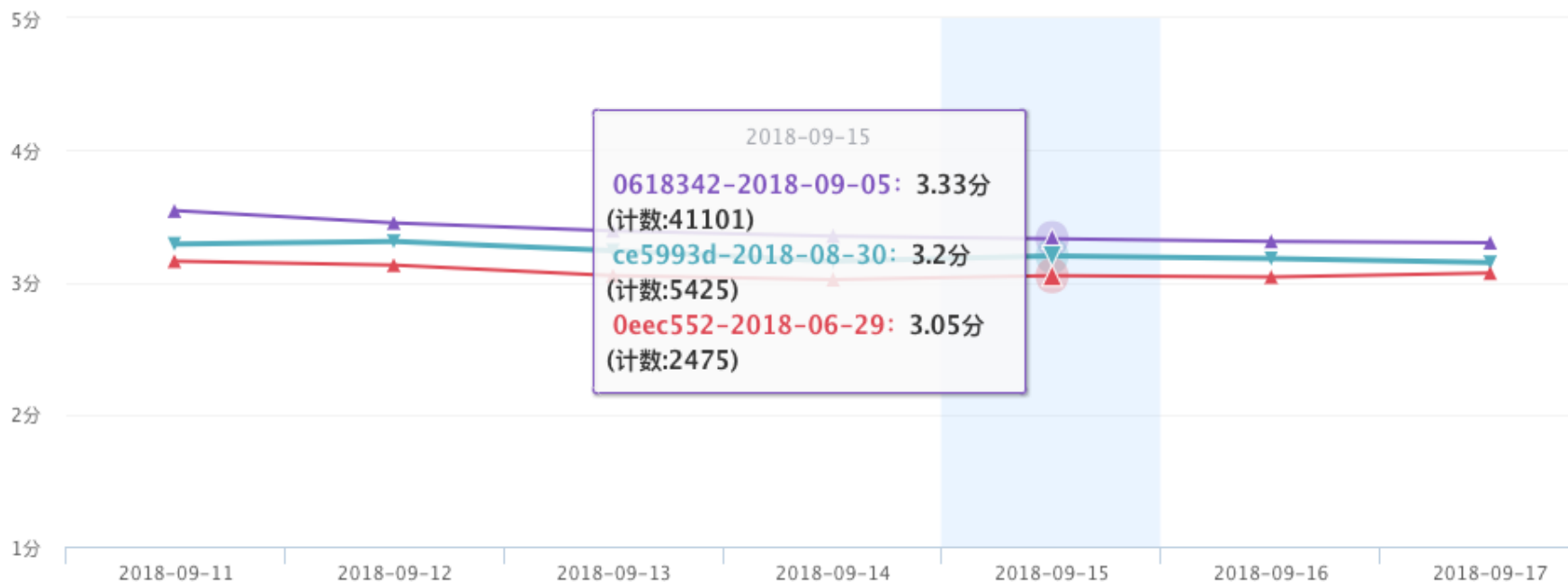
全部网络▼

全部▼

全部对端▼

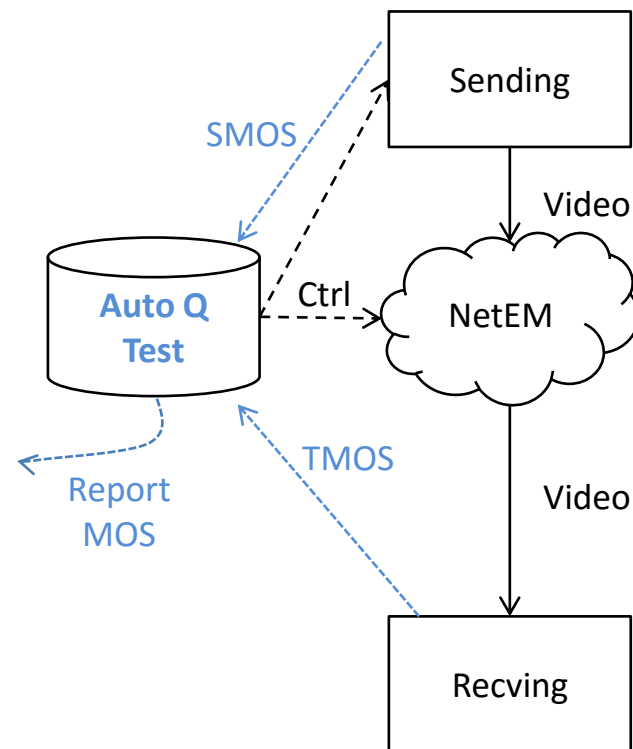
MOS▼

MOS(分) 计数391817, 均值3.25



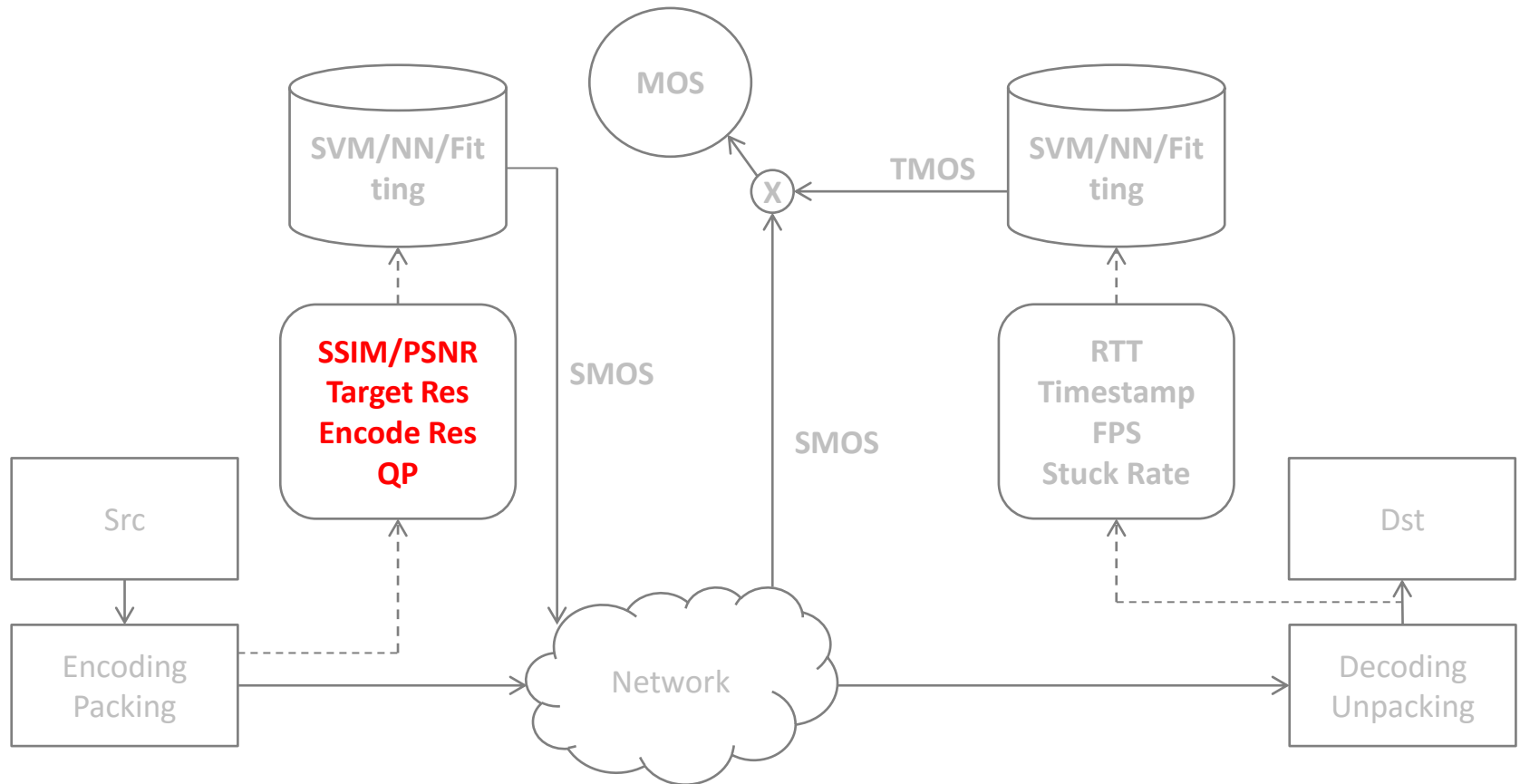
实验室应用：MOS自动评估测试

ScriptName	NetworkCondition	MOS				
2.1.8_P8_720p_bw1000k_shipin_test		v4.01	v4.01	v4.02-200	v4.02-221	v5.00
Average	[60~140] Delay=90ms jitter=30ms loss=0% Bandwidth=2000Kbps Buffer=1000 aggregation=0%	4.69	4.74	4.72	4.69	4.74
2.1.8_P8_180p_normalwifi_shipin_test						
Average	[0~60] Delay=30ms jitter=10ms loss=1% Bandwidth=800Kbps Buffer=1000 aggregation=10%	4.89	4.92	4.89	4.89	4.84
Average	[60~140] Delay=60ms jitter=30ms loss=3% Bandwidth=700Kbps Buffer=1000 aggregation=10%	4.72	4.64	4.72	4.77	4.77
Average	[140~200] Delay=100ms jitter=80ms loss=2% Bandwidth=500Kbps Buffer=1000 aggregation=10%	4.59	4.39	4.42	4.69	4.54
2.1.8_P8_180p_unstable3G_shipin_test						
Average	[0~80] Delay=100ms jitter=60ms loss=20% Bandwidth=2000Kbps Buffer=1000 aggregation=20%	4.29	4.29	4.39	4.39	4.62
Average	[80~140] Delay=200ms jitter=100ms loss=15% Bandwidth=300Kbps Buffer=1000 aggregation=85%	4.23	4.03	4.19	3.98	3.43
Average	[140~200] Delay=150ms jitter=30ms loss=0% Bandwidth=1000Kbps Buffer=1000 aggregation=10%	4.59	4.62	4.59	4.59	4.64
2.1.8_P8_180p_unstablewifi_shipin_test						
Average	[0~60] Delay=50ms jitter=10ms loss=1% Bandwidth=1000Kbps Buffer=1000 aggregation=10%	4.47	4.59	4.49	4.47	4.69
Average	[60~140] Delay=60ms jitter=30ms loss=10% Bandwidth=200Kbps Buffer=1000 aggregation=70%	3.21	3.94	3.47	3.41	3.17
Average	[140~200] Delay=30ms jitter=10ms loss=0% Bandwidth=2000Kbps Buffer=1000 aggregation=10%	4.72	4.74	4.77	4.69	4.67
Average	[200~260] Delay=150ms jitter=150ms loss=12% Bandwidth=500Kbps Buffer=1000 aggregation=75%	4.39	4.42	4.47	4.32	4.02

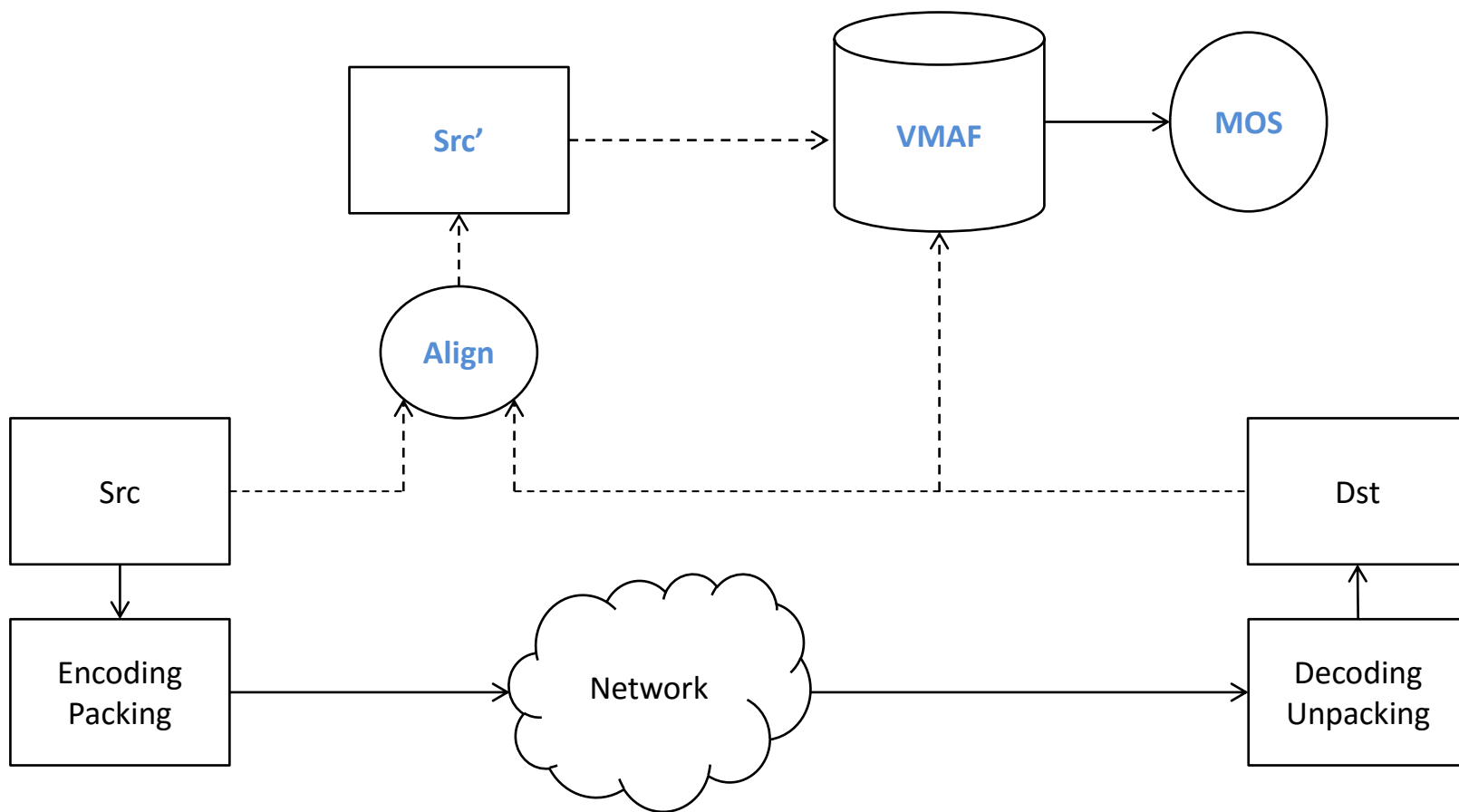


CEV 的缺点

- 依赖编码器，不同编码器需要重新训练（调参）
- 不是端对端，无法对第三方系统做打分

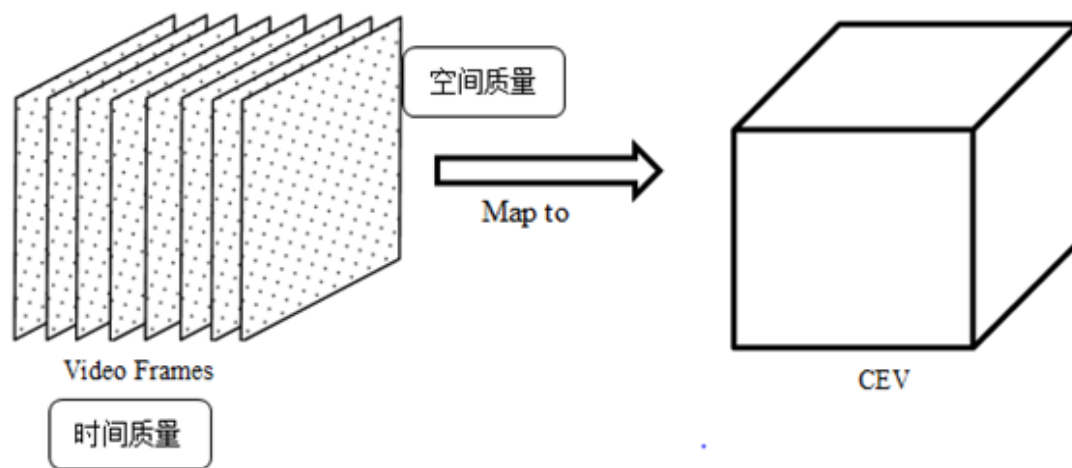


再讨论实时视频的端到端MOS评分



Part2: 动机

- 能否使用CEV应用到优化编码策略，寻找视频质量甜点（基于感知编码）？



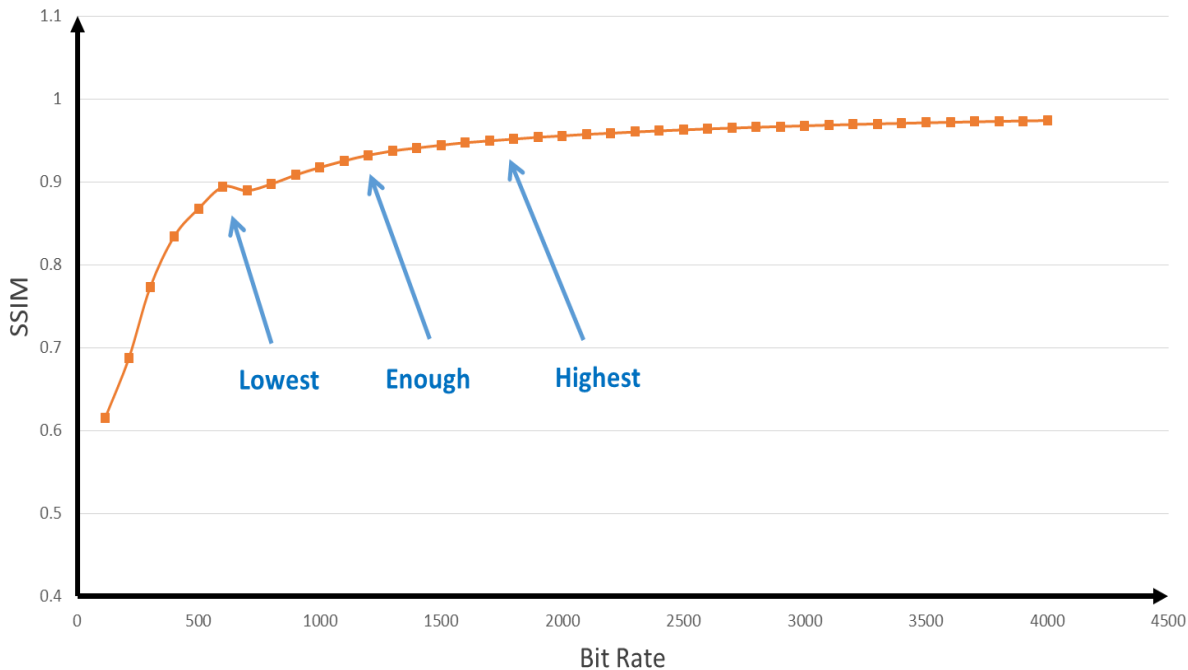
什么是甜点?

Frame Size/Frame Rate	Target Bitrate (VOD, kbps)	Min Bitrate (50%)	Max Bitrate (145%)
320x240p @ 24,25,30	150	75	218
640x360p @ 24,25,30	276	138	400
640x480p @ 24,25,30	512 (LQ), 750 (MQ)	256 (LQ) 375 (MQ)	742 (LQ) 1088 (MQ)
1280x720p @ 24,25,30	1024	512	1485
1280x720p @ 50,60	1800	900	2610
1920x1080p @ 24,25,30	1800	900	2610
1920x1080p @ 50,60	3000	1500	4350
2560x1440p @ 24,25,30	6000	3000	8700
2560x1440p @ 50,60	9000	4500	13050
3840x2160p @ 24,25,30	12000	6000	17400
3840x2160p @ 50,60	18000	9000	26100

<https://developers.google.com/media/vp9/settings/vod/>
Recommended VOD bitrates

SMOS – 编码空间质量

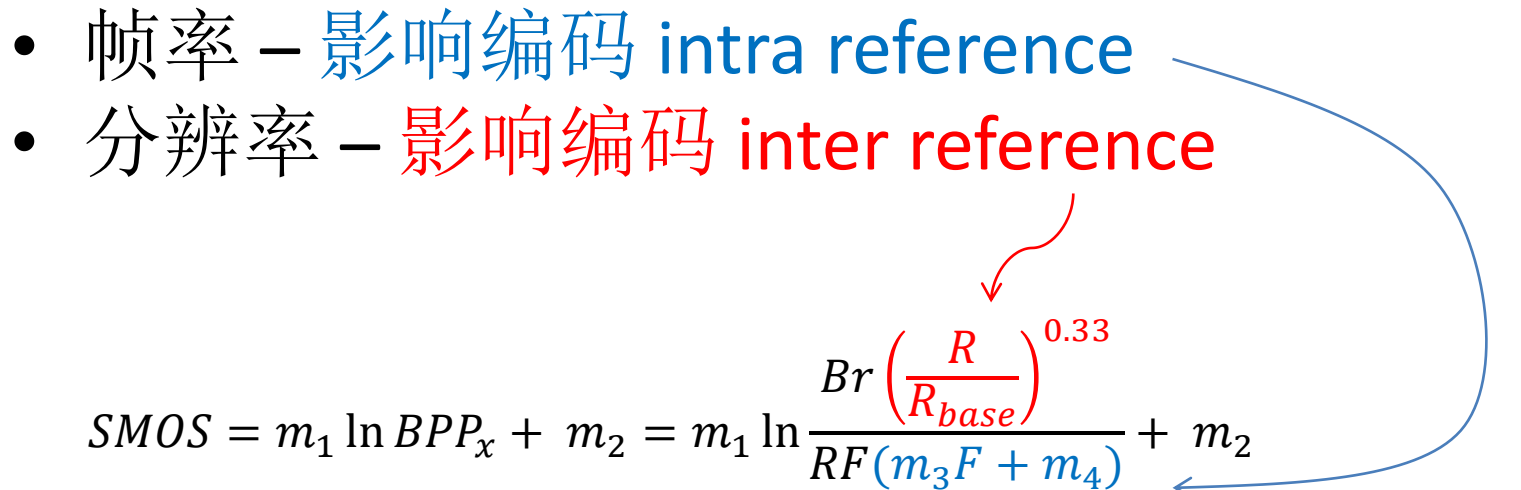
The relationship between SSIM and Bit Rate



BPP	CIF size Subjective quality
0.098	Lowest
0.195	Enough
0.296	Highest

SMOS – 分辨率、帧率关系

- 帧率 – 影响编码 intra reference
- 分辨率 – 影响编码 inter reference

$$SMOS = m_1 \ln BPP_x + m_2 = m_1 \ln \frac{Br \left(\frac{R}{R_{base}} \right)^{0.33}}{RF(m_3 F + m_4)} + m_2$$


$$BPP_x = \left(\frac{Br}{RF} \right) \alpha = BPP_{base} \alpha$$

$$\alpha = \frac{\left(\frac{R}{R_{base}} \right)^{0.33}}{m_3 F + m_4}$$

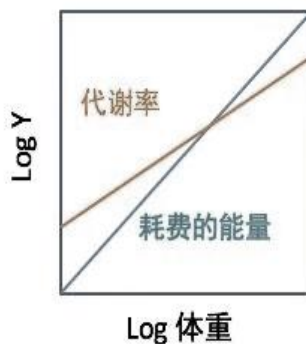
规模法则 - 杰弗里·韦斯特（Geoffrey West）

Scaling Law

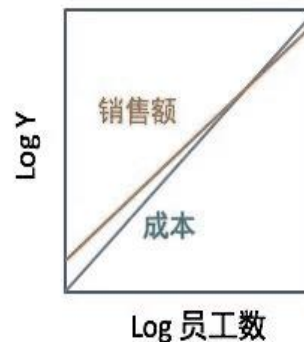
$$Y \sim X^\alpha$$

系统	生命体	公司	城市
规模尺度	体重	员工数	人口
获取的能量	代谢率	销售额	产出
消耗的能量	身体维护	成本	基础设施

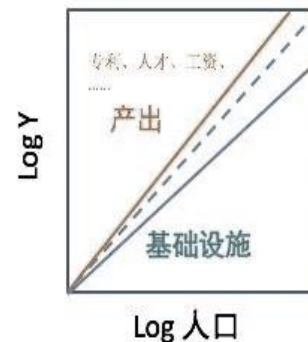
代谢率~体重^{3/4}
消耗的能量~体重



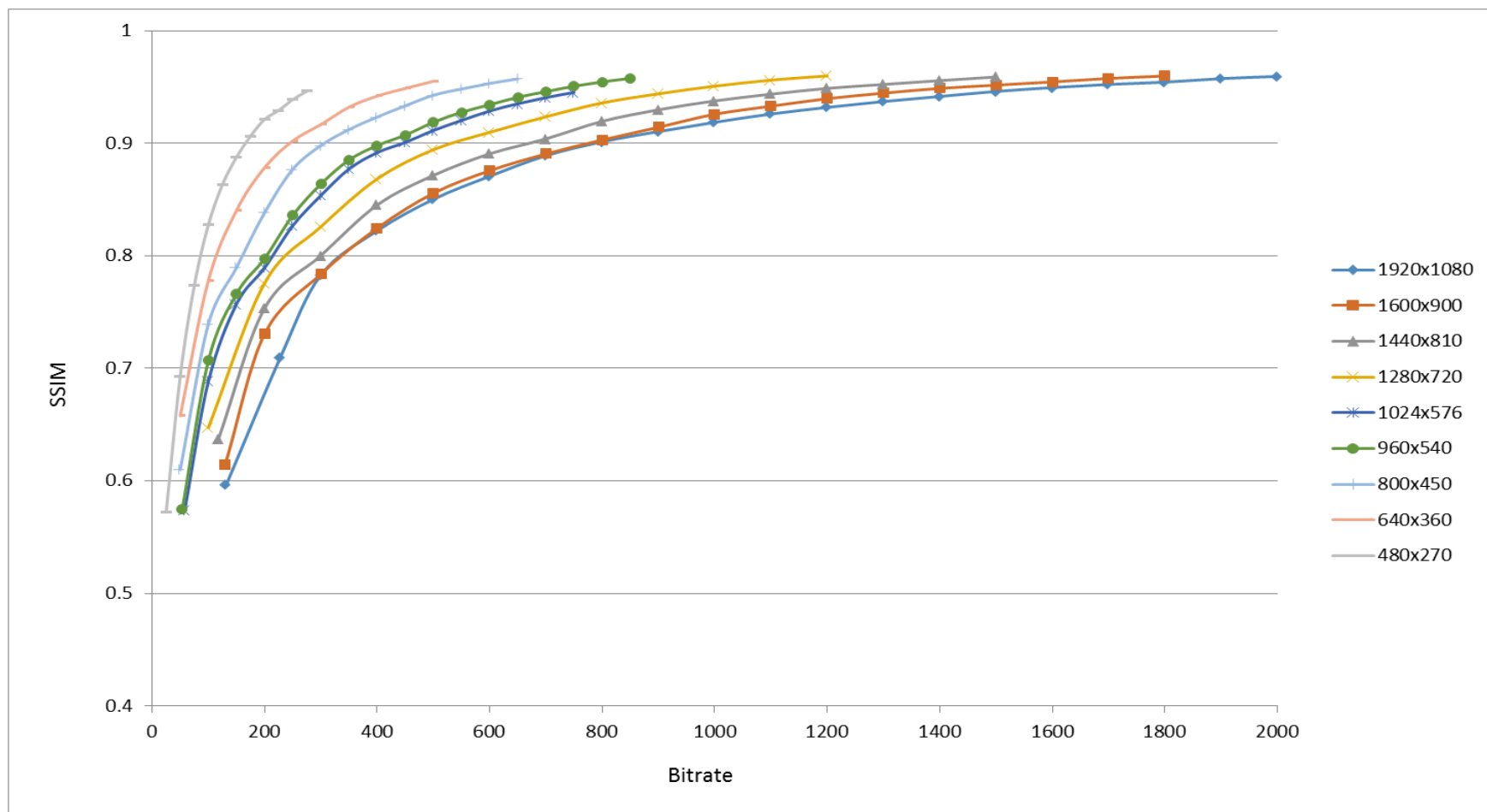
销售额~员工数^{0.9}
成本支出~员工数



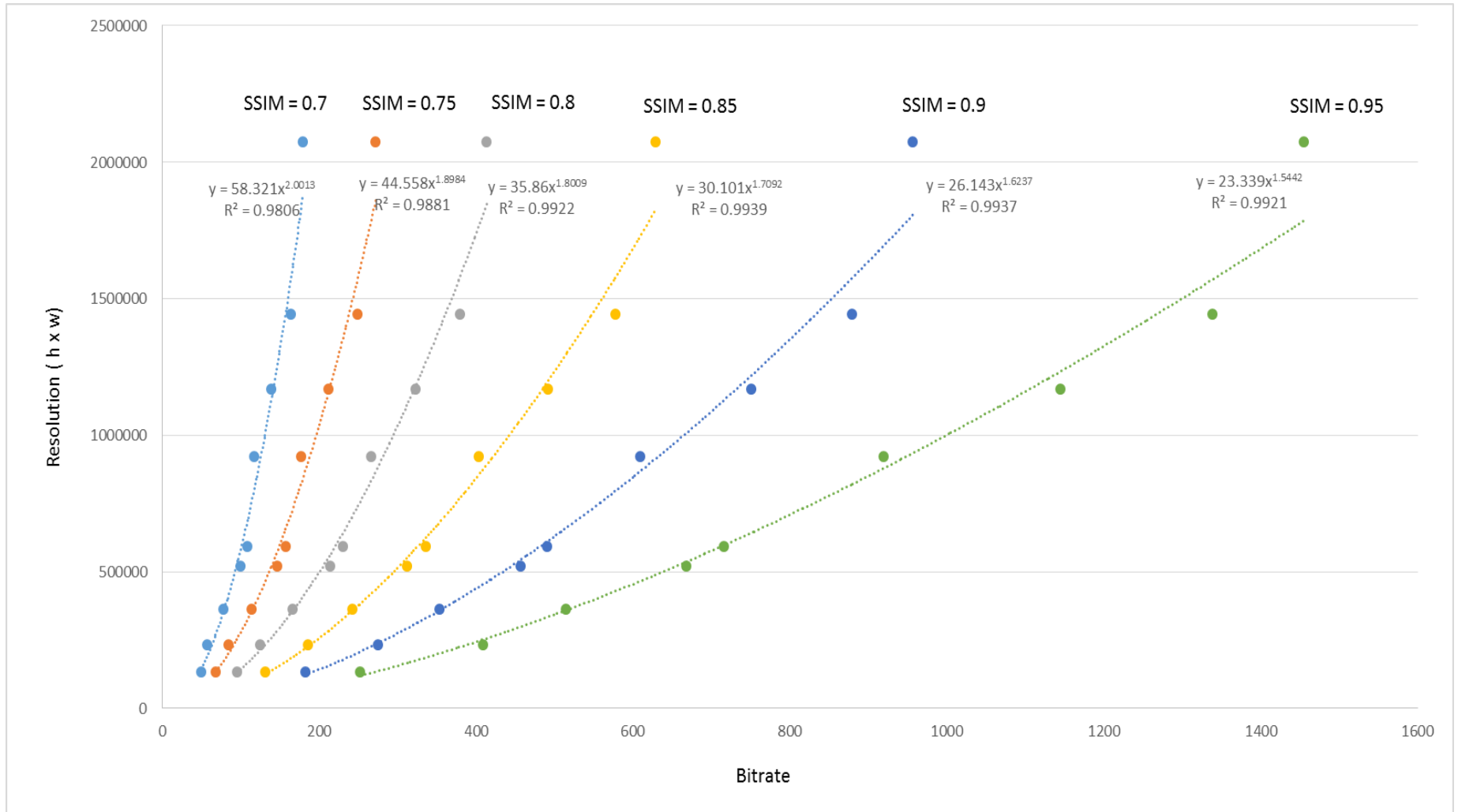
产出~人口^{1.15}
基础设施~人口^{0.85}



计算SSIM – 不同分辨率和编码码率



SMOS – 分辨率的规模法则



应用规模法则

$$\frac{R_n}{R_m} = \left(\frac{BPP_n}{BPP_m} \right)^x$$

- x 取值1.5~3
- x 越大编码器越适合高分辨率

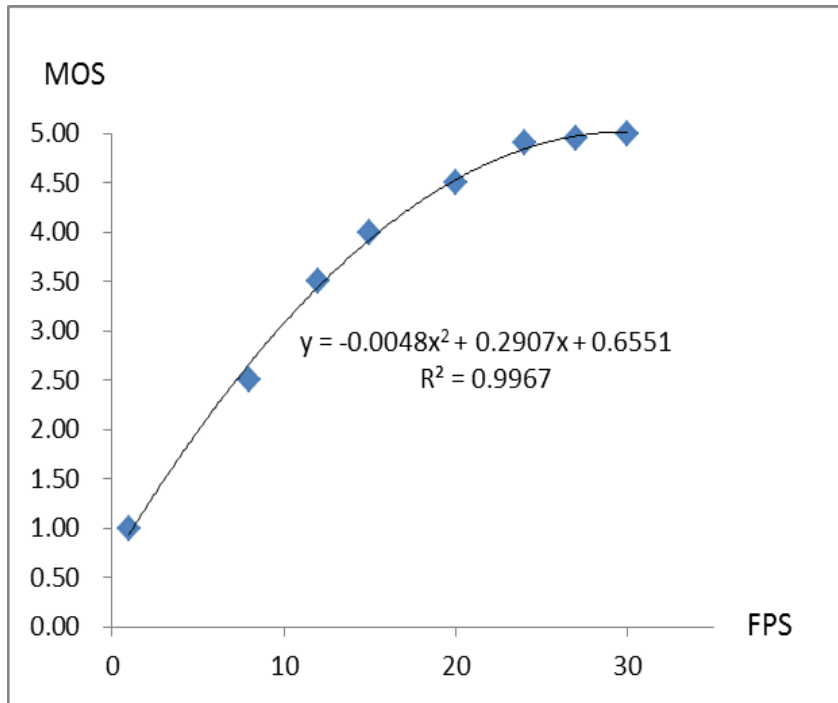
Openh264 编码举例

Resolution R_x (Height*Width)	Lowest BPP	Enough BPP	Highest BPP
100000(baseline)	0.098	0.195	0.296
200000	0.0796	0.1584	0.2404
300000	0.0705	0.1402	0.2129
400000	0.0647	0.1287	0.1953
500000	0.0605	0.1203	0.1826
600000	0.0573	0.1139	0.1729
700000	0.0547	0.1088	0.1651
800000	0.0525	0.1045	0.1586
900000	0.0507	0.1009	0.1531

$$BPP_x = BPP_{base} \left(\frac{R_x}{R_{base}} \right)^{0.33}$$

- 不同编码器参数不一样
- 不同场景也不一样（适用于有一定时空复杂度的视频）

TMOS -- 编码时间质量



$$TMOS = m_5 F^2 + m_6 F + m_7$$

F is frame-rate, $m_5 = -0.0048$, $m_6 = 0.2907$, $m_7 = 0.6651$.

CEV MOS - 编码总体质量

$$CEV = SMOS^{\alpha} * TMOS$$

$$CEV = \left[m_1 \ln \frac{Br \left(\frac{R}{R_{base}} \right)^{0.33}}{RF(m_3 F + m_4)} + m_2 \right]^{\alpha} (m_5 F^2 + m_6 F + m_7)$$

$$0.296 > BPP = \frac{Br \left(\frac{R}{R_{base}} \right)^{0.33}}{RF(m_3 F + m_4)} > 0.02$$

求导 - 求极值

$$\frac{dCEV}{dF} = \alpha \left[m_1 \ln \frac{Br \left(\frac{R}{R_{base}} \right)^{0.33}}{R * F(F * m_3 + m_4)} + m_2 \right]^{\alpha-1} \frac{-m_1(2m_3 + m_4)}{F(m_3F + m_4)} (m_5F^2 + m_6F$$

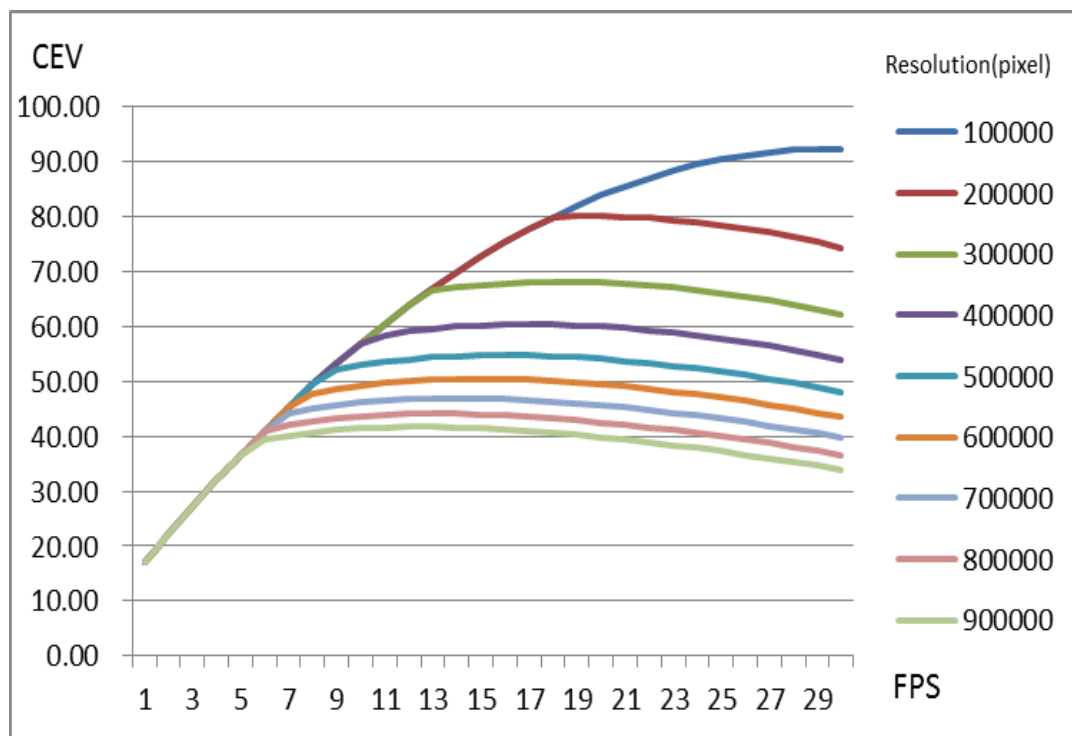
$$+ m_7) + \left[m_1 \ln \frac{Br \left(\frac{R}{R_{base}} \right)^{0.33}}{R * F(F * m_3 + m_4)} + m_2 \right]^{\alpha} (2m_5F + m_6) = 0$$

求解

$$Br = \frac{e^{\left[\alpha \frac{(2m_3+m_4)(m_5F^2+m_6F+m_7)}{F(m_3F+m_4)(2m_5F+m_6)} - \frac{m_2}{m_1} \right]}}{\left(\frac{R}{R_{base}} \right)^{0.33}} RF(m_3F + m_4)$$

$$0.296 > e^{\left[\alpha \frac{(2m_3+m_4)(m_5F^2+m_6F+m_7)}{F(m_3F+m_4)(2m_5F+m_6)} - \frac{m_2}{m_1} \right]} = BPP = \frac{Br \left(\frac{R}{R_{base}} \right)^{0.33}}{RF(m_3F + m_4)} > 0.02$$

举例：应用甜点模型



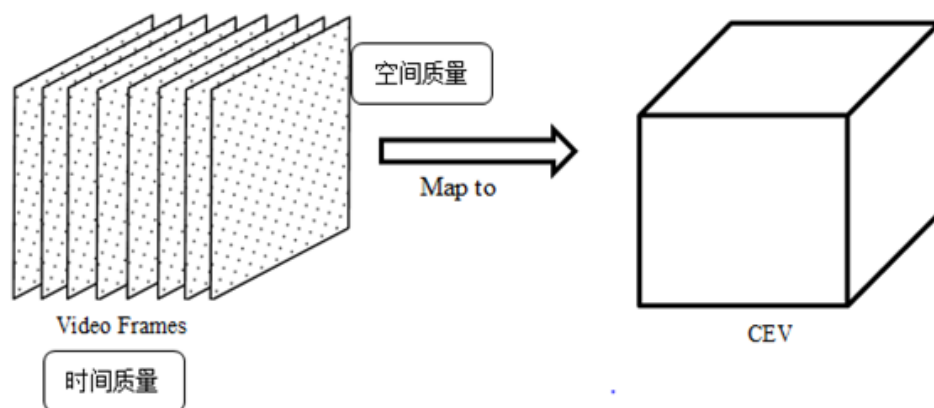
- 给定800kbps带宽，求解不同分辨率下最佳帧率
- 基线值，编码器不同、视频复杂度不同时需要调整

Resolution	Video sequence sample	AVC kbps	best frame-rate choice						
			avgFPS	Obser1	Obser2	Obser3	Obser4	Obser5	CEV
176*144	ANSI_SRC_3inrow	10	6.33	5	4	9	8	10	2
		30	6.50	9	7	5	6	8	4
	ANSI_SRC_boblec	20	7.67	10	5	9	9	10	3
		50	8.50	13	9	8	8	5	8
	CBC_SRC_LePoint	40	3.67	3	3	4	3	3	6
		80	7.83	7	7	8	6	7	12
	ANSI_SRC_washdc	100	17.00	16	18	15	15	20	18
		150	21.17	21	24	18	25	20	19
352*288	Sign_Irene	50	7.67	12	6	8	9	9	2
		100	11.67	15	10	12	12	15	6
	ANSI_SRC_5row1	150	11.33	14	13	10	10	8	13
		350	20.33	17	25	10	25	25	20
	FlamingoHilton	300	16.50	8	11	22	22	22	14
		500	22.00	14	24	25	25	25	19
	Cheerleaders	400	17.00	13	15	20	20	18	16
		200	7.33	6	6	6	7	7	12
640*480	Redflower	100	10.67	14	15	10	8	10	7
		200	13.33	18	24	10	8	10	10
	BetesPasBetes	500	15.00	17	25	10	15	10	13
		400	12.67	14	20	10	10	10	12
	CaesarsPalace	600	17.67	16	25	20	15	15	15
		1200	23.00	23	25	18	25	25	22
	ITU_SRC_Football	700	21.67	18	25	25	25	22	15
		1400	25.00	25	25	25	25	25	25
720*576	itur625_30mobileandcale ndar	300	13.50	18	15	12	12	12	12
		600	18.67	25	25	18	15	12	17
	itur625_15flowergarden	1000	22.33	25	25	20	25	18	21
		1800	24.17	25	25	20	25	25	25
1280*720	FighterPilot	600	13.33	15	15	15	15	15	5
		400	13.50	15	15	15	12	15	9
	ducks_take_off	1500	9.83	7	8	10	10	10	14
		1800	11.67	9	10	13	11	13	14
Pearson coefficient				0.833	0.895	0.871	0.946	0.899	0.874

对比测试：CEV预测的配置跟人工调校优化配置一致

总结

- CEV 是一种有效的质量评估模型
- 可用于实时视频通信质量评价
- 也可用于质量甜点算法模型设计



Thank you

