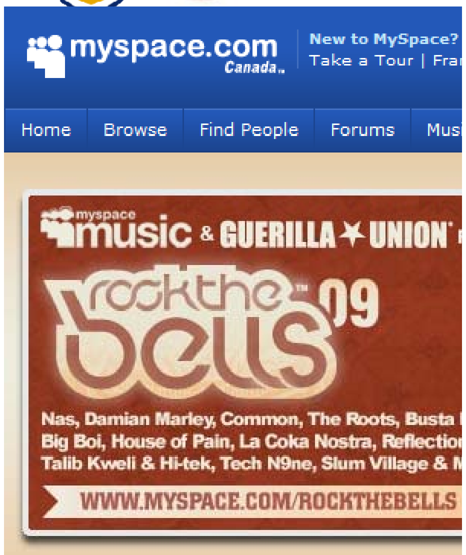
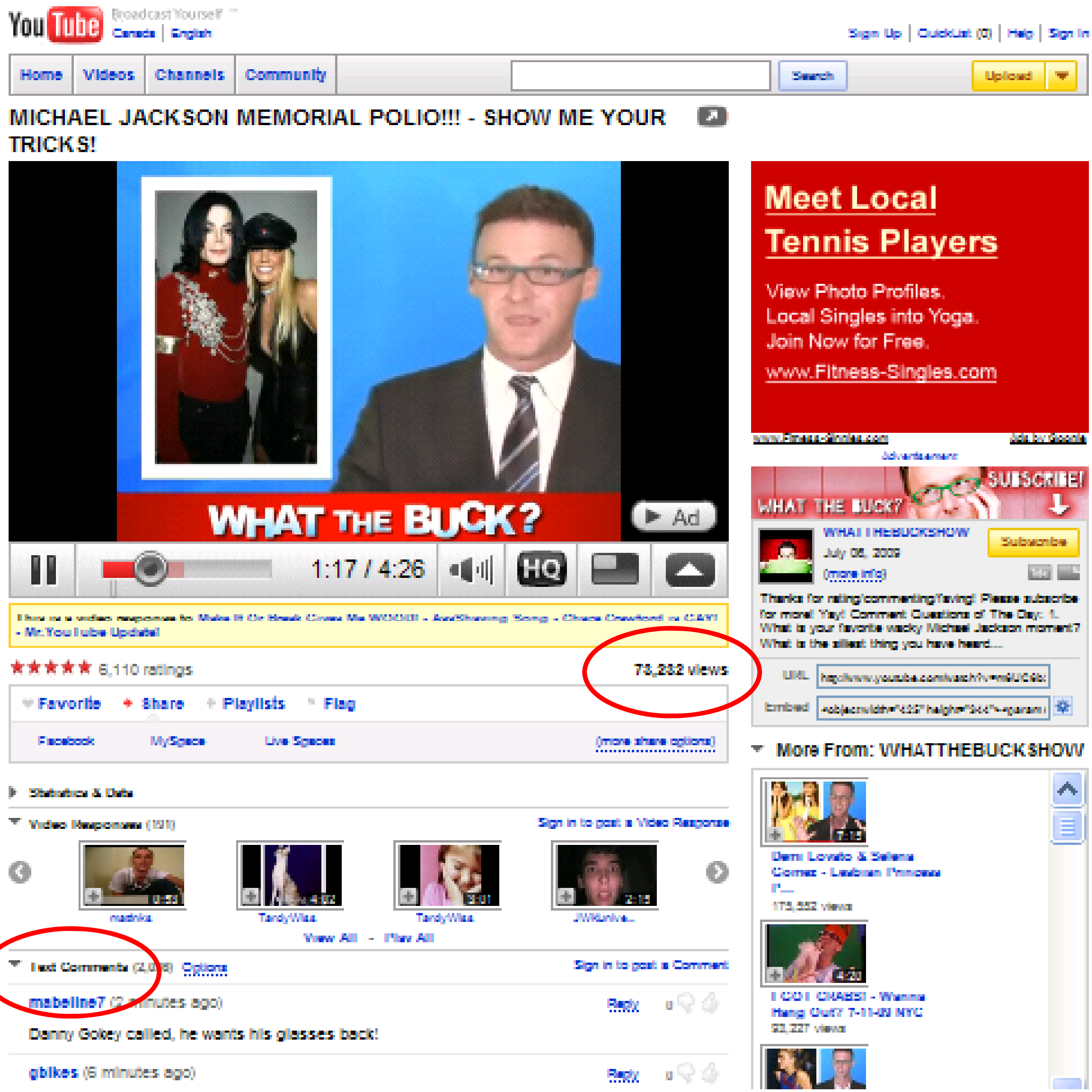
Information Management and Security in Media-Sharing Social Networks

Rabab Ward, Z. Jane Wang, Mehrdad Fatourechi,
ECE Department,
University of British Columbia (UBC), Vancouver, Canada

H. Vicky Zhao
ECE Department,
University of Alberta, Edmonton, Canada

Acknowledgement: Xudong Lv, Ehsan Nezhadarya, Amir Va lizadeh, and Mani Malekesmaeili from UBC, Binglai Niu and Bo Hu from University of Alberta





Favourite among teens

Social Media

- Definition from Wikepedia:
  - Social media is online content created by people using highly accessible and scalable publishing technologies.
  - Social media is a shift in how people discover, read and share news, information, content and media etc.
  - It's a fusion of sociology and technology, transforming monologues (one to many) into dialogues (many to many).

- Social media can take many different forms (e.g. Internet forums, weblogs, social blogs, wikis, podcasts, pictures and video). There are different types of social media applications,
  - e.g. communication (e.g. blogs, social networking), multimedia, entertainment, collaboration (e.g. Wikis), news/opinion etc.

Social Media (cont'd)

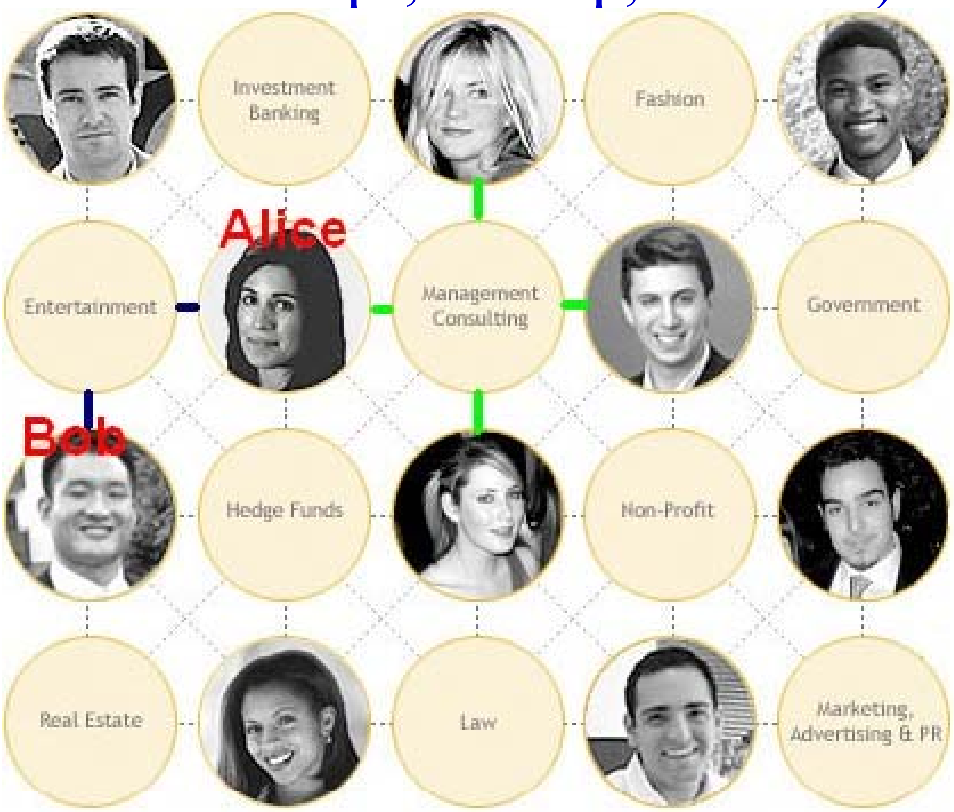
- Examples of multimedia social media applications
  - Photo sharing: Flickr, Zooomr, Photobucket, SmugMug
  - Video sharing: YouTube, Vimeo,
  - Livecasting: Ustream.tv, Justin.tv, Stickam, bizbuzztour.com
  - Audio/Music Sharing: imeem, The Hype Machine, Last.fm, ccMixter

- Examples of social networking app.

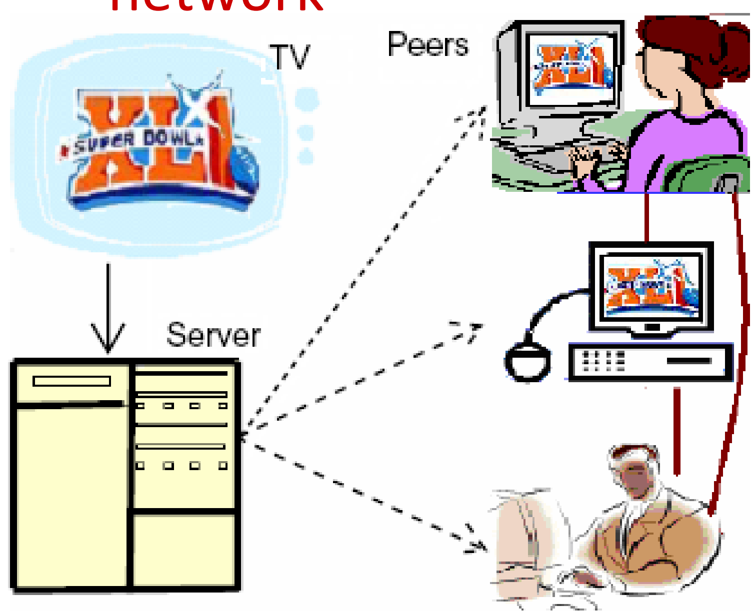
Bebo, Facebook, YouTube, LinkedIn, MySpace, Orkut, Skyrock, Hi5, Ning, Elgg, Google Groups, Twitter, etc.

Social Networks

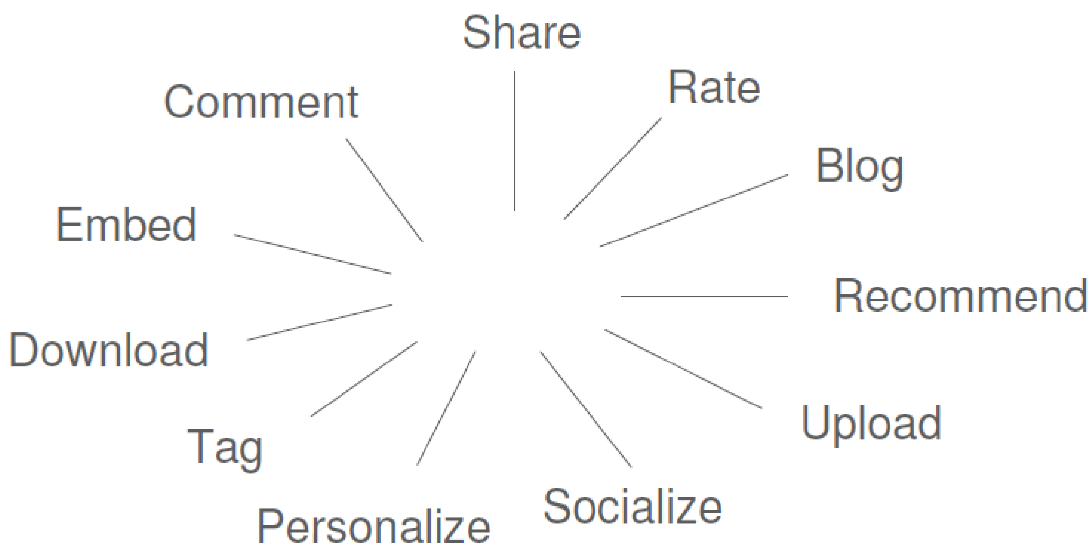
- A social network is an on-line social structure made of nodes tied by some types of interdependency (relations, e.g. values, friendships, kinship, trade etc)



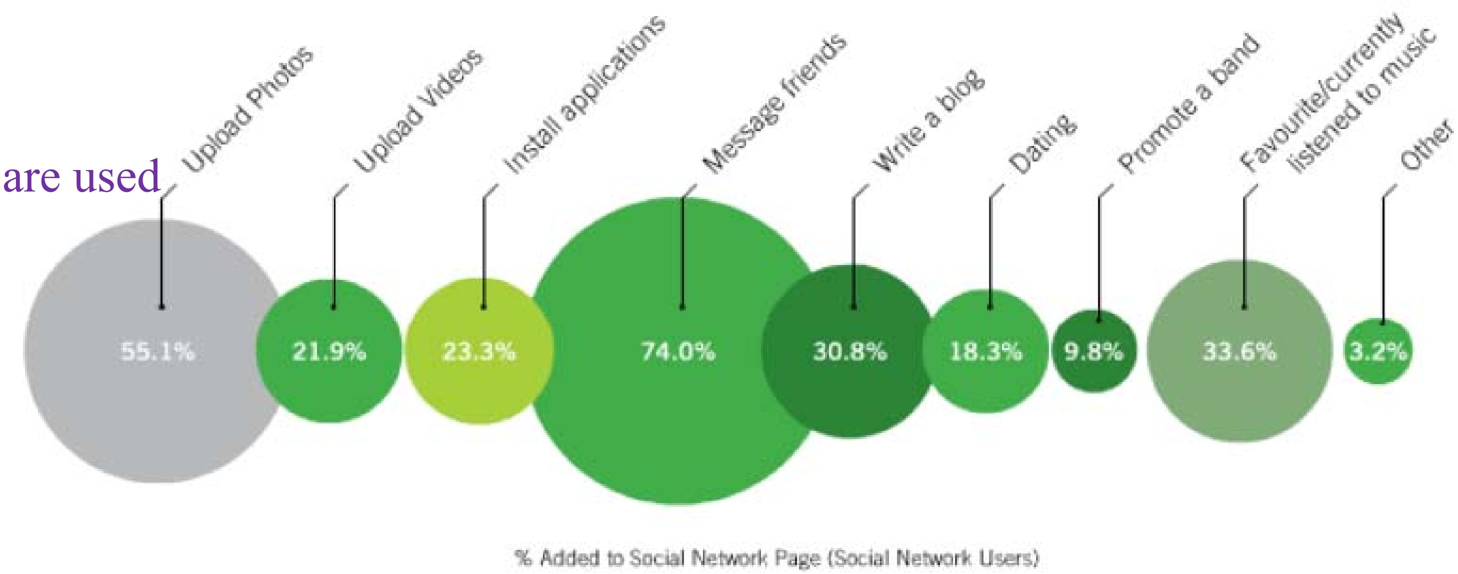
P2P live streaming social network



Social networking motivate to:



Social networks are used for:

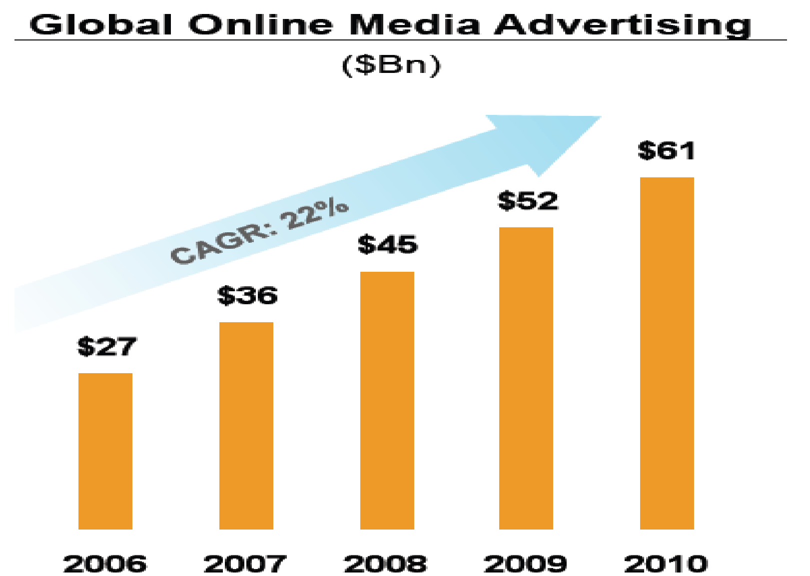


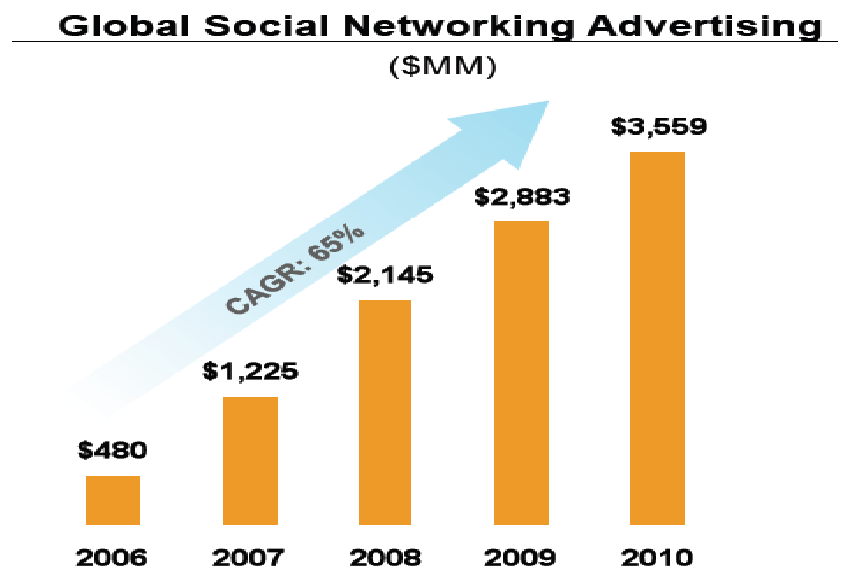
From http://edidaktik.tgm.ac.at/fachtagung08/trampedach_intro-to-social-networking.pdf

Business vision:

Alexa Global Traffic ranking in 2008

	2008
1	YAHOO!
2	YouTube
3	Google
4	Windows Live Beta
5	msn
6	myspace a place for friends
7	facebook
8	hi5
9	Wikipedia.org
10	orkut



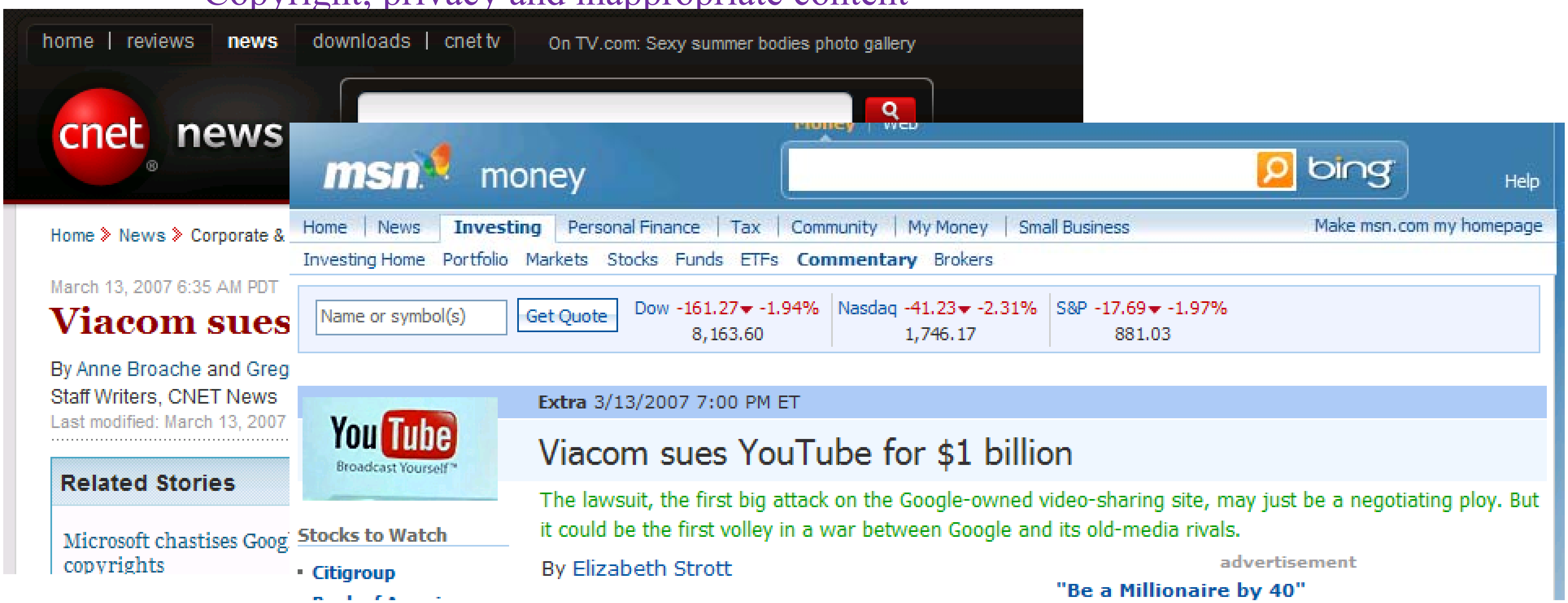


Source: eMarketer

Social media, social networking, social whatever ….

YouTube

- YouTube (social networking and video sharing): a video sharing website on which users can upload and share videos.
- Social impact: YouTube (launched in 2005) made it possible, simple for ordinary computer users to post on-line videos that millions of people could watch, and turned video sharing into one important Internet culture.
- However there are major criticism of YouTube:
  - Copyright; privacy and inappropriate content



Technical Challenges

- Security issues: content protection and DRM

TABLE 1 USTR "SPECIAL 301" PIRACY LOSS ESTIMATES FOR COPYRIGHT INDUSTRIES - 2005 [a]

Loss Estimates for Selected Countries Only [b]

U.S. Industry	Piracy Loss Asia/Pacific ($ Millions)	Piracy Loss Europe/The CIS ($ Millions)	Piracy Loss The Americas ($ Millions)	Piracy Loss Middle East/Africa ($ Millions)
Motion Pictures	$593.0	$1,014.0	$1,120.0	$186.0
Recorded Music	$710.8	$773.9	$1,133.3	$86.7
Business Software	$3,476.0	$3,086.4	$1,493.0	$583.0
Entertainment Software	$1,357.6	$1,021.1	$258.5	$15.6
Sub-Total	**$6,137.4**	**$5,895.4**	**$4,004.8**	**$871.3**
		Total Losses All Regions	**$16,908.9**	

[a] Source: *International Intellectual Property Alliance*, USTR 2007 "Special 310" Decisions, May 1, 2007.
[b] These estimates do not include losses incurred in the United States, United Kingdom, France, Germany, Australia and a number of other countries.

Source: S. Siwek, "The True Cost of Copyright Industry Piracy to the U.S. Economy", IPI Policy Report, Oct. 2007

Technical Challenges(cont'd)

- Reliability and scalability:
  - *January 28, 2006, pplive broadcasted the annual Chinese Spring Festival Gala to over 200K users at bit rate 400-800 kbps (100 gigabits per second)*
- Media management: e.g. multimedia indexing and content recognition

SONG TITLE	NUMBER OF VERSIONS	NUMBER OF COPIES	α
Naughty Girl	26,715	631,387	0.80672
Ocean Avenue	8,000	174,106	0.80339
Where is the Love ?	48,613	448,987	1.0215
Hey Ya	46,926	734,108	0.86035
Toxic	38,992	650,529	0.86135
Tipsy	32,893	853,688	0.77721
My Band	49,447	1,816,663	0.82019

Data collected:
May 1, 2004

Source: J. Liang, etc., "Pollution in P2P File Sharing Systems", IEEE InfoCom, 2005

Outline

To address these challenges, we investigate fundamental technologies including

- Extrinsic forensic watermarking
- Intrinsic multimedia forensics
- Content-based fingerprinting for media content recognition
  - FJLT image hashing algorithms
  - Automation of image hashing algorithms
  - Video hashing

and at the system-design level,

- Behavior modeling and analysis
- Automated network-service (quality) monitoring
  - Watermarking-based quality monitoring
  - Hash-based quality monitoring

Automated quality monitoring methods

- Watermarking-based quality monitoring
- Hash-based quality monitoring

Image Quality Assessment via Image Hashing

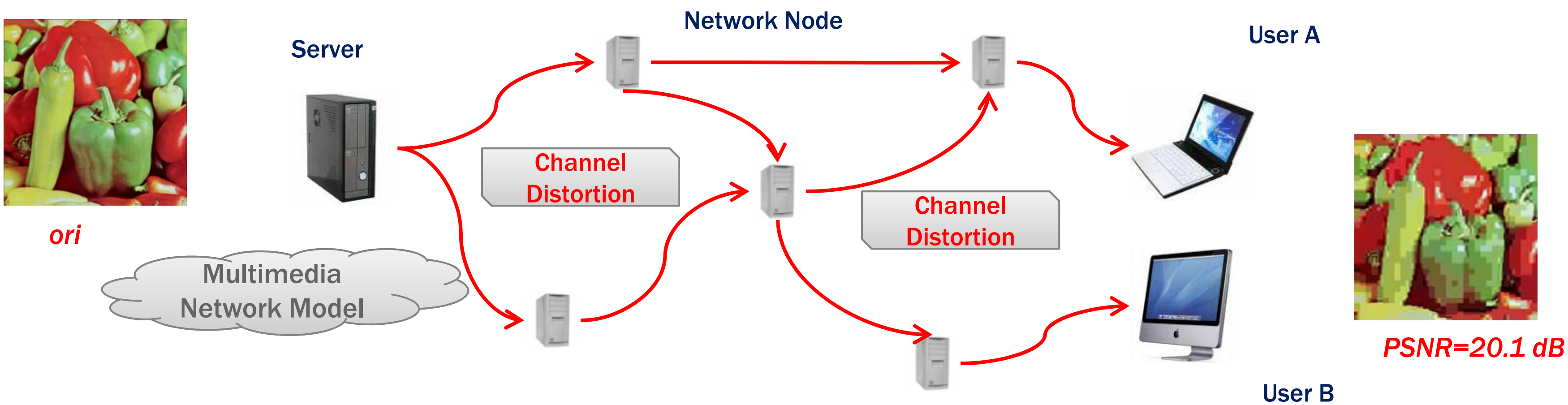


Image Quality Assessment Approaches:

- Full-reference quality assessment: Compare original image and distorted copies

  Pros: accurate assessment; *Cons*: original image is not available in practice.

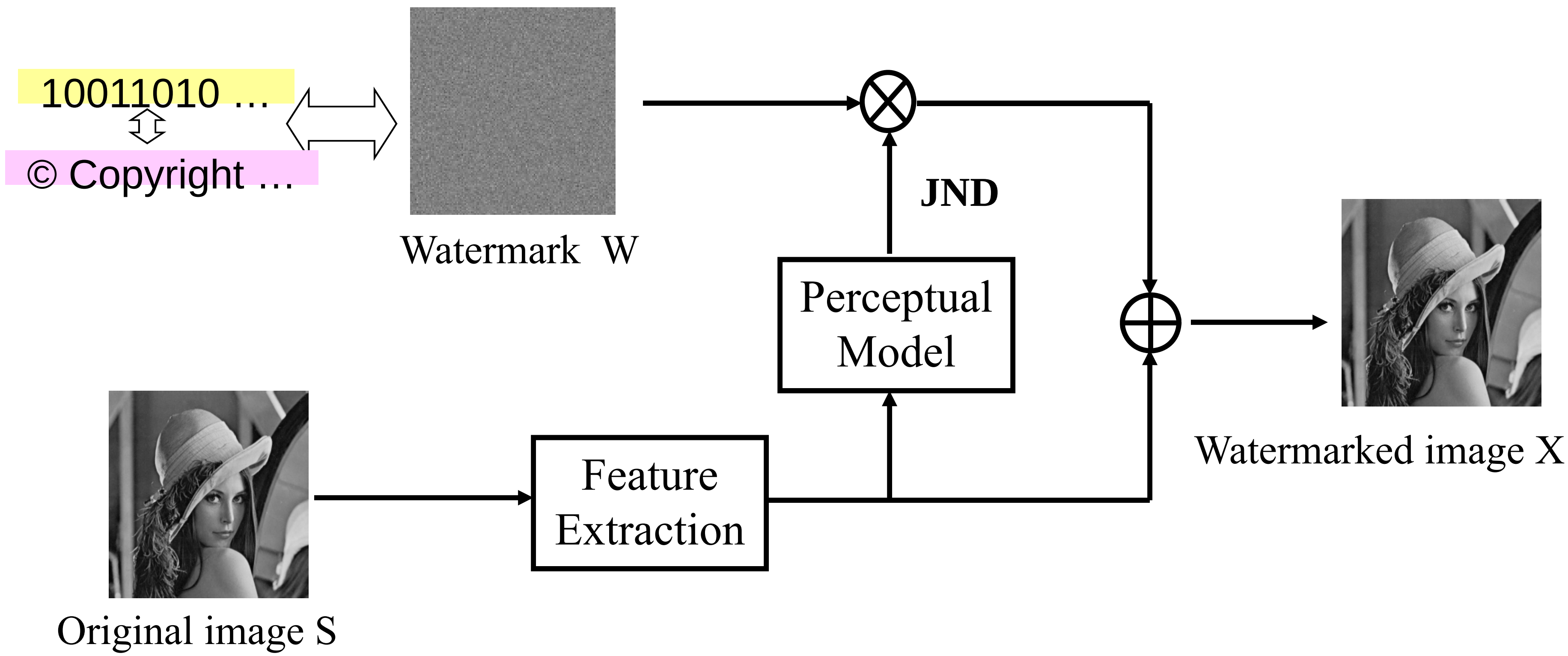
- No-reference quality assessment: Assess image quality without any information of original image

  Pros: easy to assess; *Cons*: not accurate.

- Reduced-reference quality assessment: A trade-off between FR and NR scheme. Require a *partial information* of original image

  Pros: applicable in practice; *Cons*: accuracy depends on the information

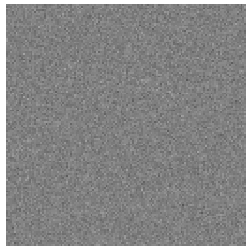
Robust Watermark Embedding



- Additive spread spectrum embedding: $\boxed{X = S + JND \cdot W}$
  - The noise-like watermark is spread all over the entire host signal
  - JND: just-noticeable-difference from human visual models

Correlation based WM detection for QA

Watermark W

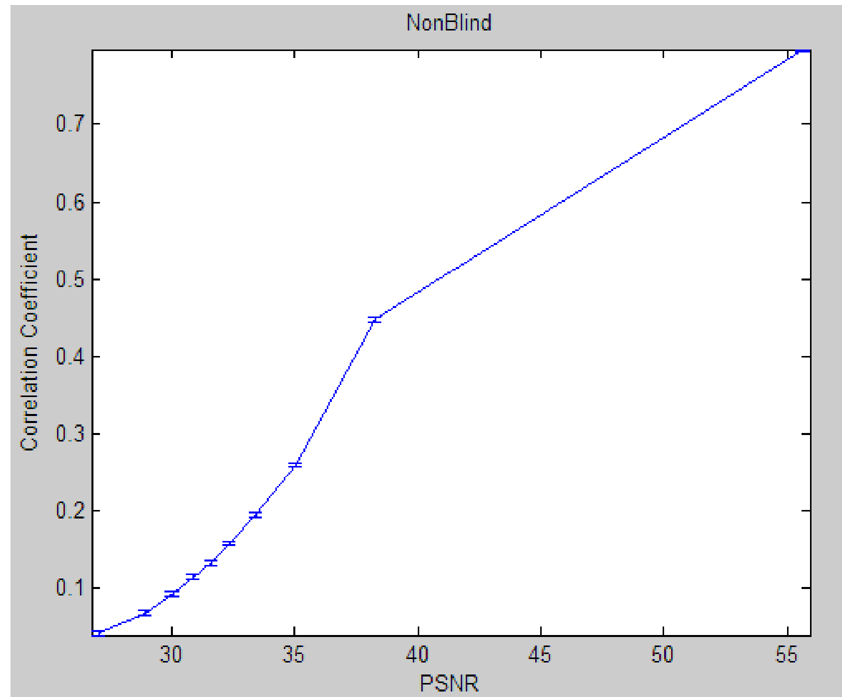


Received image Y

Watermark detection

Detection statistic R

The LOD R vs. PSNR quality curve



locally optimum detector (LOD):

$$LOD(Y^{(w)}) = \frac{\alpha}{K_1 \cdot K_2} \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} w[k_1,k_2]\, sign(Y^{(w)}[k_1,k_2]) \left| \frac{C_{T_0}[k_1,k_2]}{Y^{(w)}[k_1,k_2]} \right|^{1-\varepsilon} \left(c[k_1,k_2] \left| \beta[k_1,k_2] Y^{(w)}[k_1,k_2] \right|^{c[k_1,k_2]} - \varepsilon \right)$$

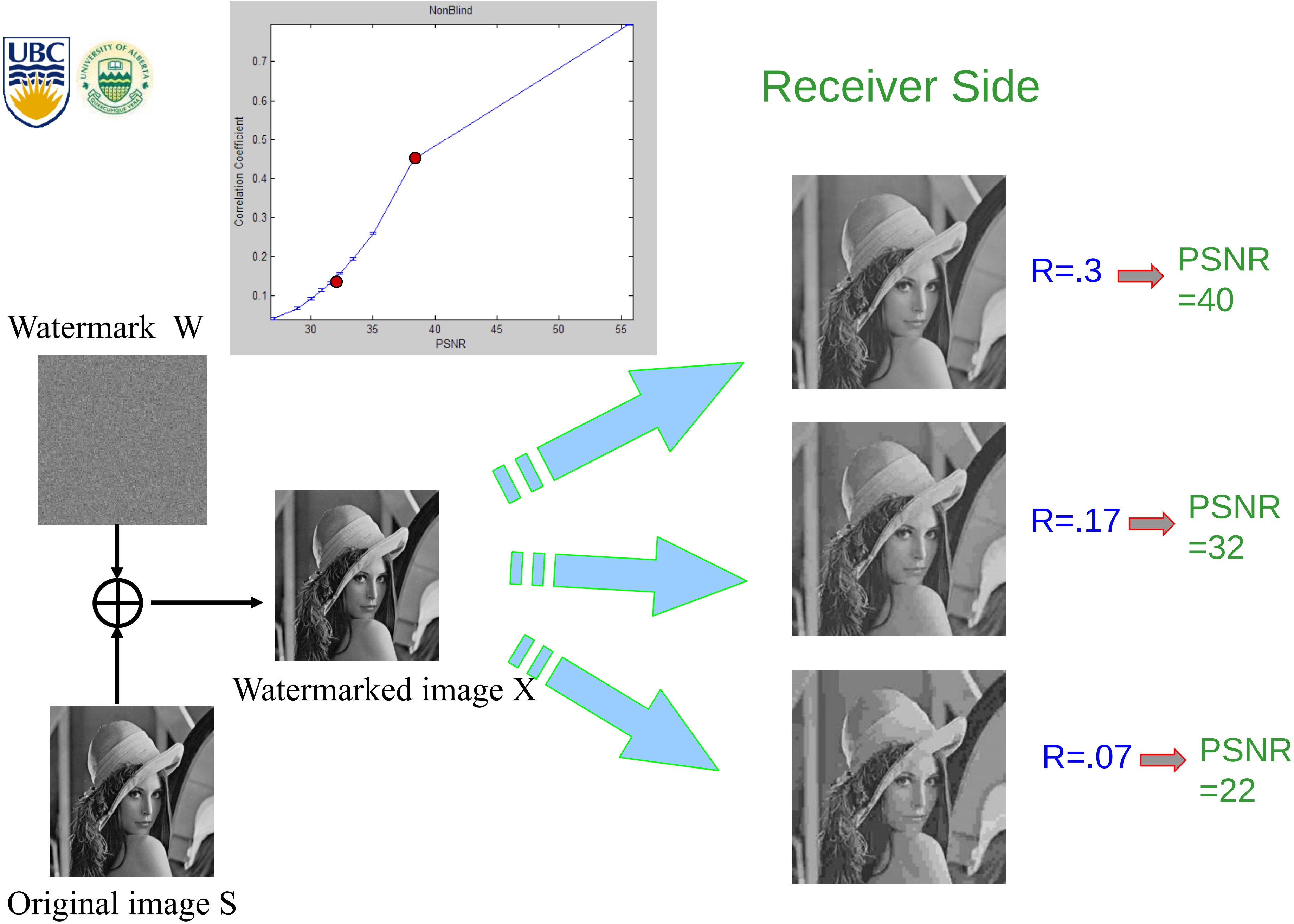

NonBlind
Correlation Coefficient
PSNR
Receiver Side
Watermark W
Watermarked image X
Original image S
R=.3
PSNR =40
R=.17
PSNR =32
R=.07
PSNR =22

Image Quality Assessment Example

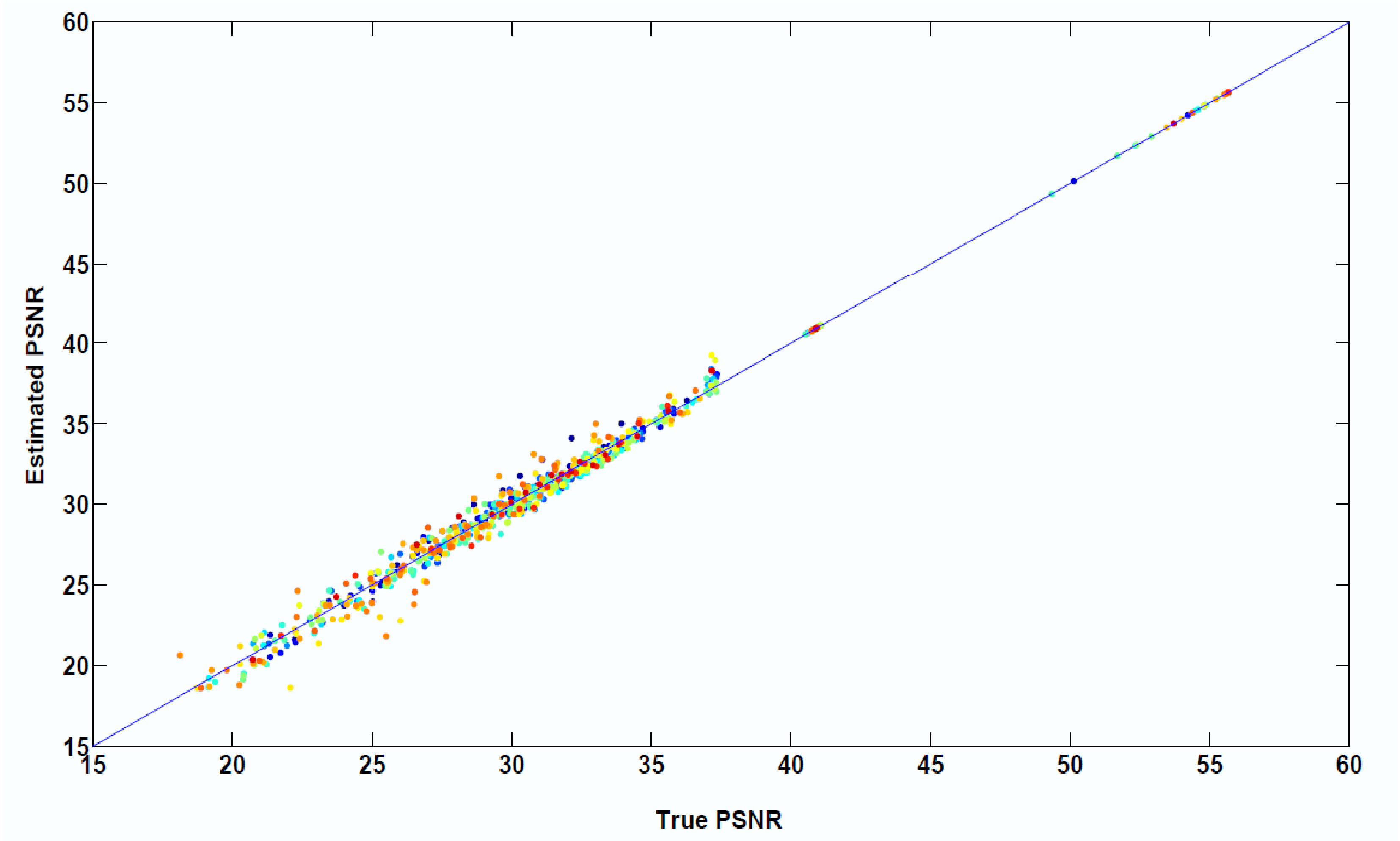
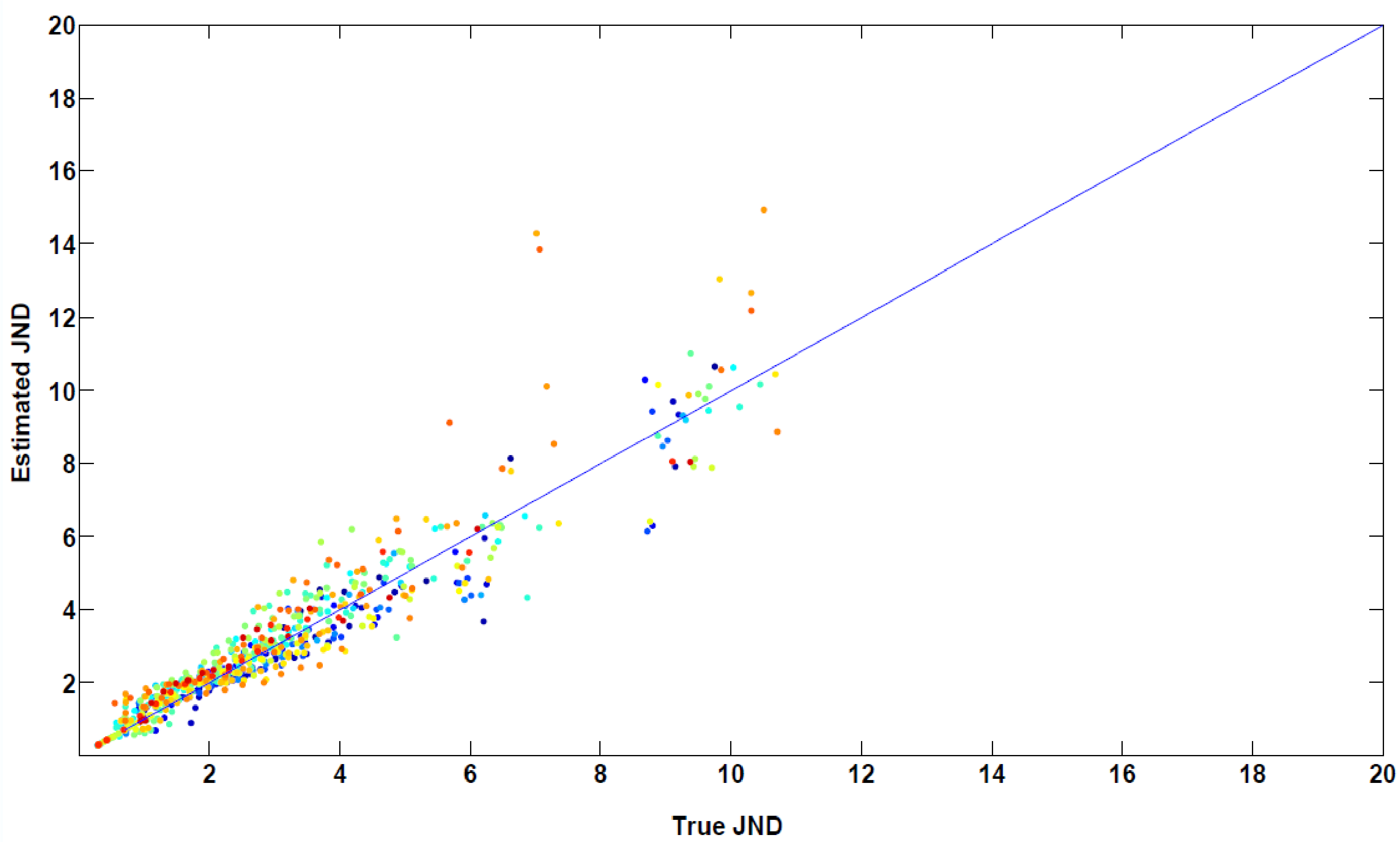
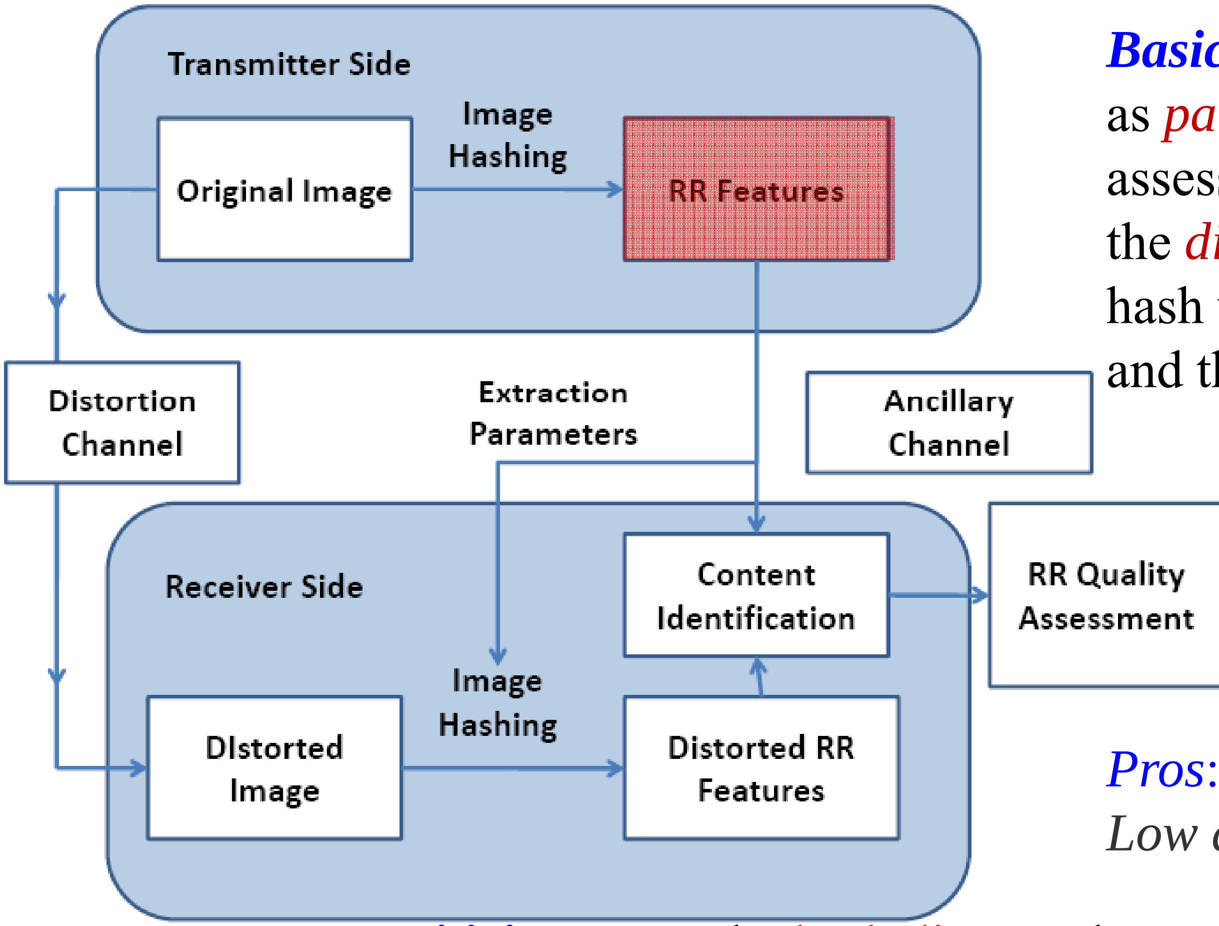


Image Quality Assessment Example



Reduced-Reference Image Quality Assessment using Image Hashing



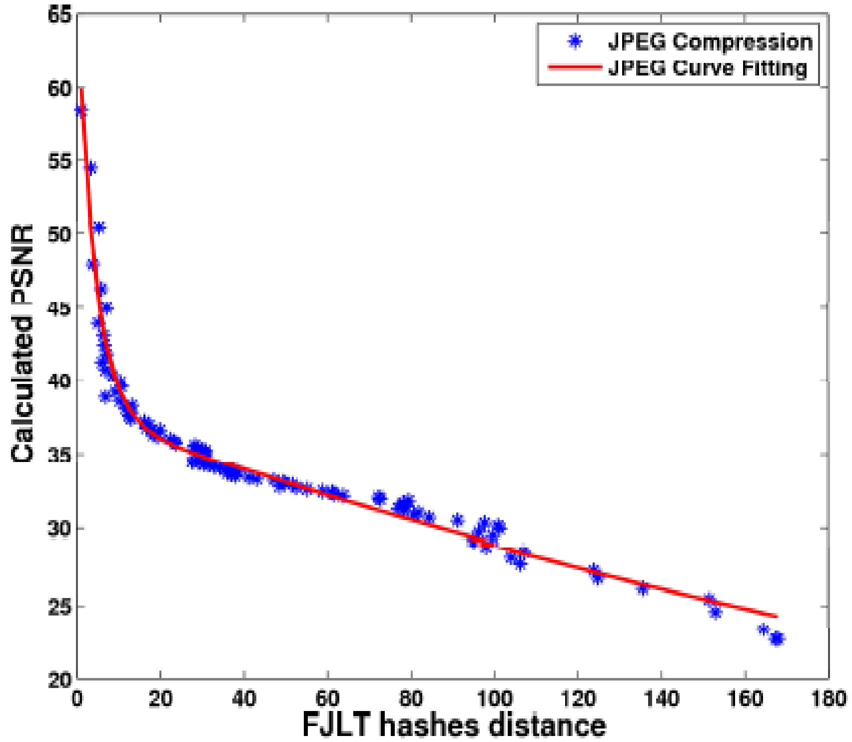
***Basic Idea*:** Using image hashes as *partial information* and assessing image quality based on the *distance* between the image hash vectors of the original image and the received image

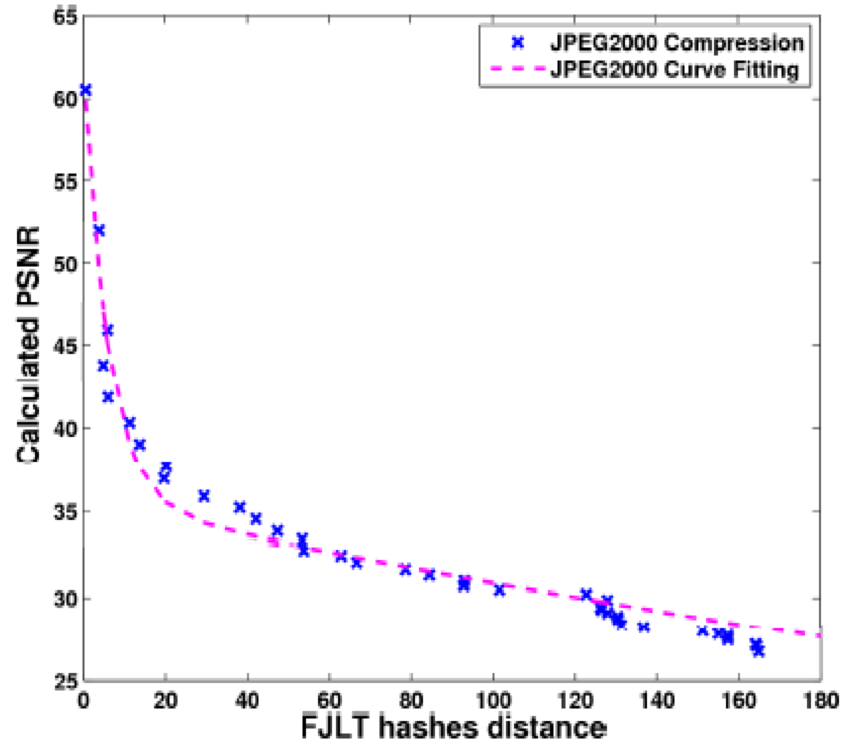
Pros:
Low data rate of RR features.

Monotone sensitivity means the hash distance between original image and its distorted copy is getting further and further when the quality of distorted image becomes worse (e.g. *PSNR is smaller*).

Monotone Sensitivity

RR Features:
*5 index + 5 original hashes + 4 curve coefficients =**14** !!! (**short enough**)*

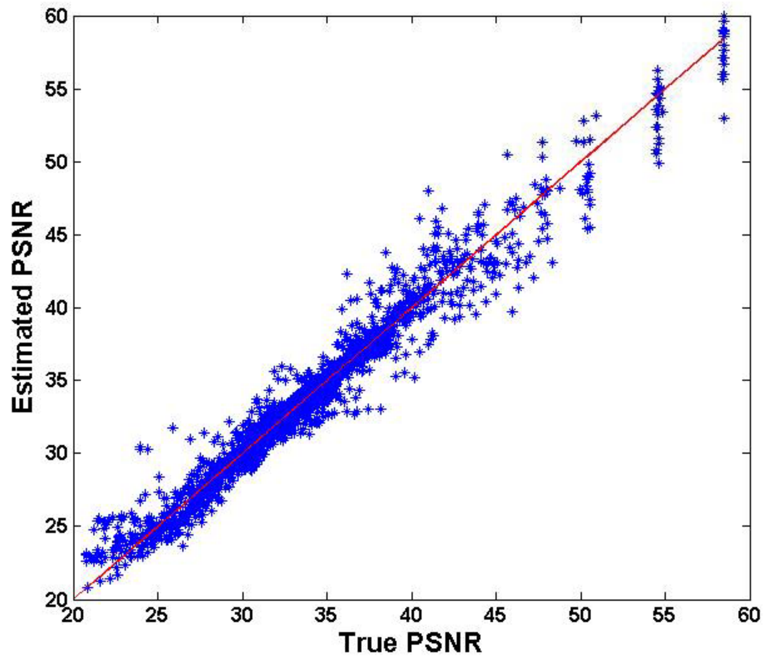




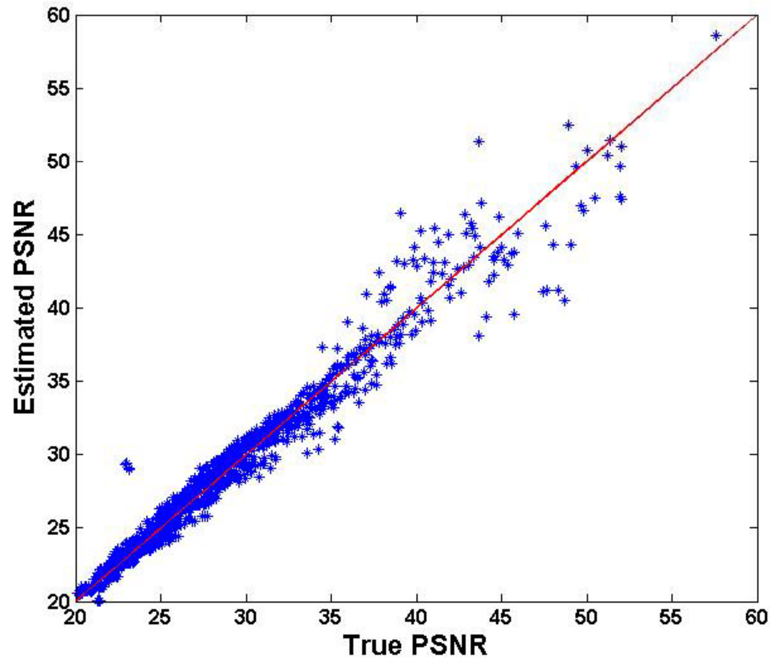
Ideally: Need an one-to-one mapping relation between hash distances and PSNR

Solution: Instead of examining the overall distance between two hashes, we investigate individual components' distances and choose the ones whose distances are monotonically sensitive to the quality degradation (PSNR)

Quality Assessment Results Under Compression



Correlation under JPEG



Correlation under JPEG2000

Correlation between estimated PSNR and true PSNR under compression

Conclusion: The FJLT hashing-based RR quality assessment, though require only a low data rate, have good perceptual relevance and thus can provide an accurate image quality estimate.

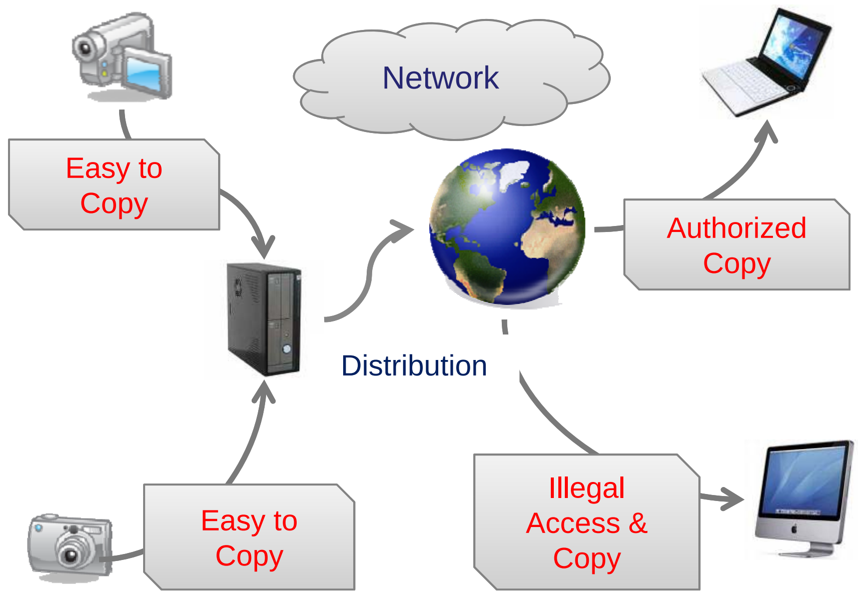
Summary

- Watermark and image hash are shown to be promising partial information for image quality assessment
- Upgrade the method to video quality assessment
- Investigate other more robust WM/hashing/etc methods
- Investigate faster and automated design methods

Content-Based Image Fingerprinting (Hashing)

- FJLT-based image hashing algorithms
- Automation of image hashing algorithms
- Video hashing

Introduction- Multimedia Fingerprinting



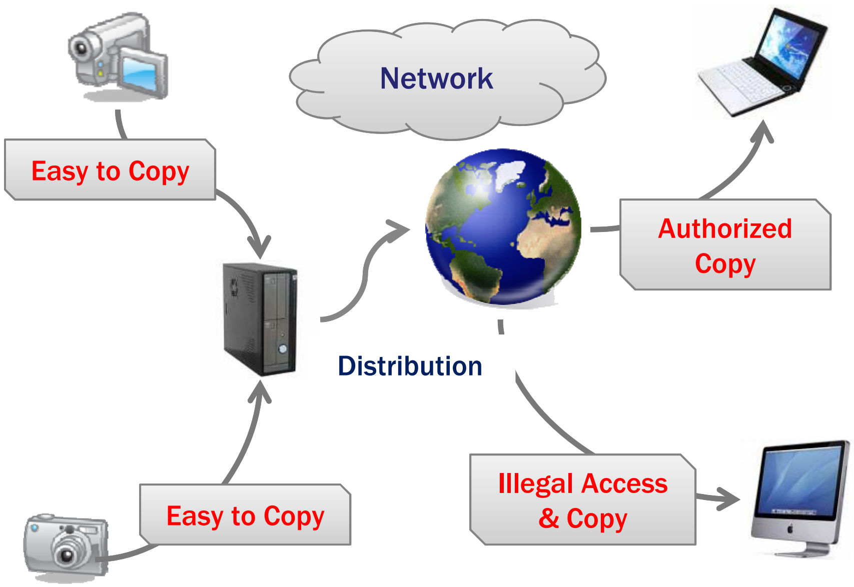
Problem: Easy-to-copy nature of digital multimedia

Q1: **How to identify efficiently from such abundant data?**

A1: Manually annotate each multimedia file with a unique descriptor in text, which could be used to index, search and identify, e.g. YouTube, Google Image.

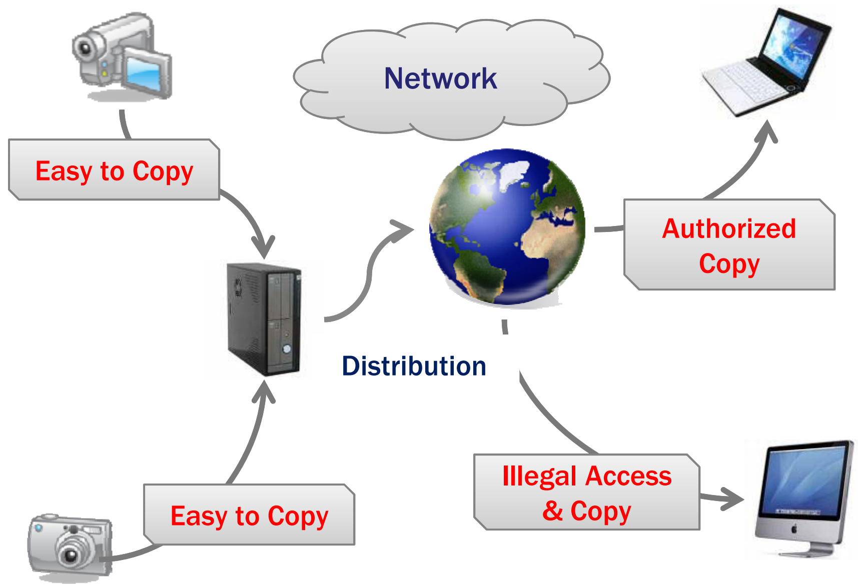
Cons: Time Consuming, inaccurate

Introduction- Multimedia Fingerprinting



Problem: Easy-to-copy nature of digital multimedia
Q2: **How to prevent unauthorized access to multimedia in terms of copyrights and protect the benefits of owners?**
A2: Watermarking (embedding identifiers into images as a proof of copyright)
Cons: Affects image quality

Introduction- Multimedia Fingerprinting



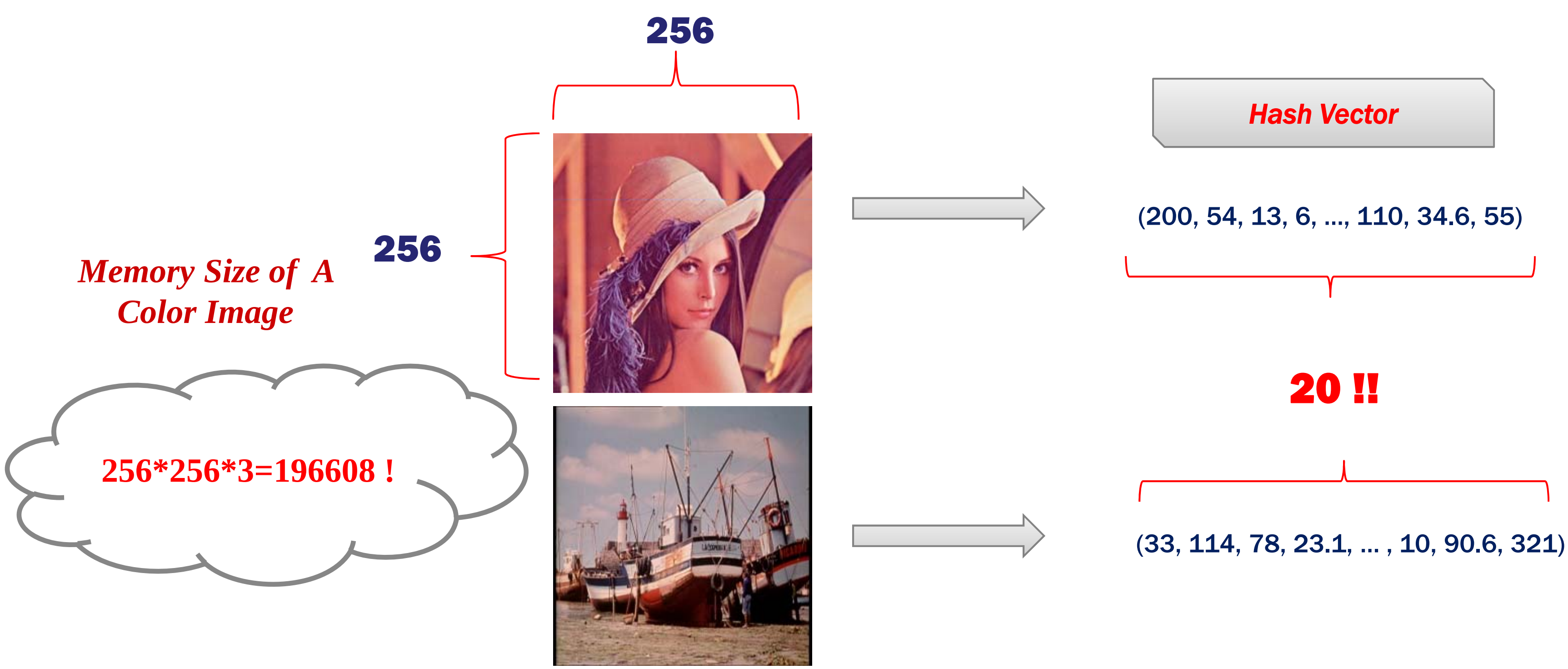
Problem: Easy-to-copy nature of digital multimedia

Solution: Implement multimedia hashing algorithms that

1. Realize fast multimedia indexing, searching, and identification (*automated description*)
2. Realize effective copyright detection and protection (*robustness and security*)

What is Image Hashing?

- Image hashing is the process of generating a short content-based digital signature (image hash) for a specific image.
- ***Advantage***: convenient storing, fast searching, and easy matching

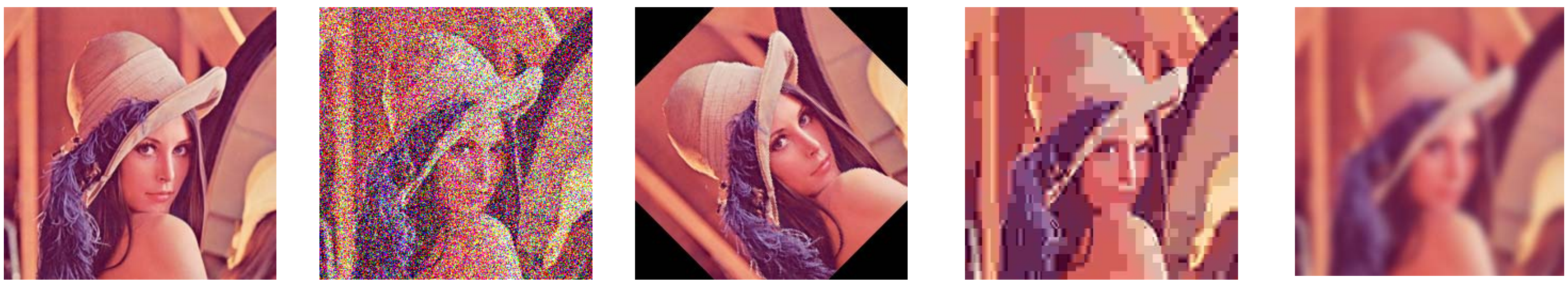


What is Image Hashing?

Critical Properties:

- Perceptual Robustness : ***content-identical images have similar hashes***

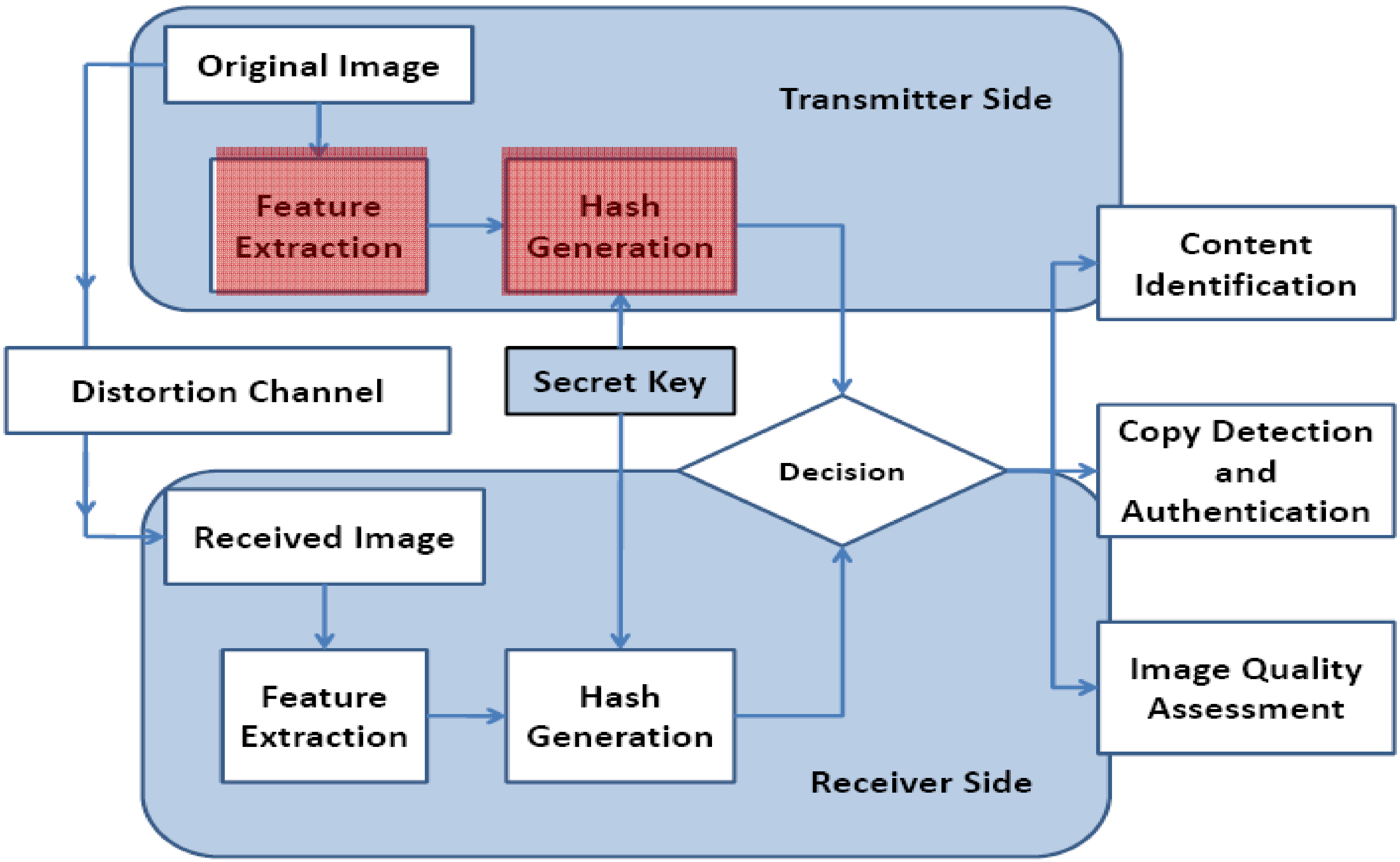
Ideally, no matter what manipulations are performed on the same image, the distorted versions have ***identical*** hashes



	Original	Gaussian Noise	Rotation	JPEG	Gaussian Blurring
Hash Vector	(100, 20, 5)	(100.5, 21, 5.3)	(101, 19.1, 4.9)	(99.2, 19.7, 5.01)	(99, 20.1, 5.5)

- Security : ***prevent unauthorized access***
- Image hash generation is a pseudorandom process depending on *a secret key*

How Image Hashing Works?



Feature Extraction

- *Image Statistics*

e.g.: Image Histogram, DCT & Wavelet Coefficients

- *Perceptually Salient Points*

e.g.: Scale Invariant Feature Transform (SIFT)

- *Rotation Invariant Transform*

e.g.: Radon Transform, Fourier-Mellin Transform

- *Dimension Reduction*

e.g.: Singular Value Decomposition (SVD),
Non-negative Matrix Factorization (NMF)

Fast Johnson-Lindenstrauss Transform

- **JL lemma**: ***P***roject from the original ***d*** dimensions down to a lower ***k*** dimensions while incurring a distortion of at most $\pm\varepsilon$ in their pairwise distance

X_1 (200, 54, 13, 6, ..., 110, 34.6, 55) → (100, 20, ..., 5) ΦX_1

distance

$\|X\|_2 = \|X_1 - X_2\|_2$

d **>>** ***k***

distance

$(1-\varepsilon)k\|X\|_2 \leq \|\Phi X\|_2 \leq (1+\varepsilon)k\|X\|_2$

X_2 (33, 114, 78, 23.1, ... , 10, 90.6, 321) → (23, 99, ..., 13) ΦX_2

- **Fast Johnson-Lindenstrauss Transform**:

$$\Phi = P \cdot H \cdot D, \ \ with\ size\ k \times d$$

a *k*-by-*d* matrix with elements from a normal distribution

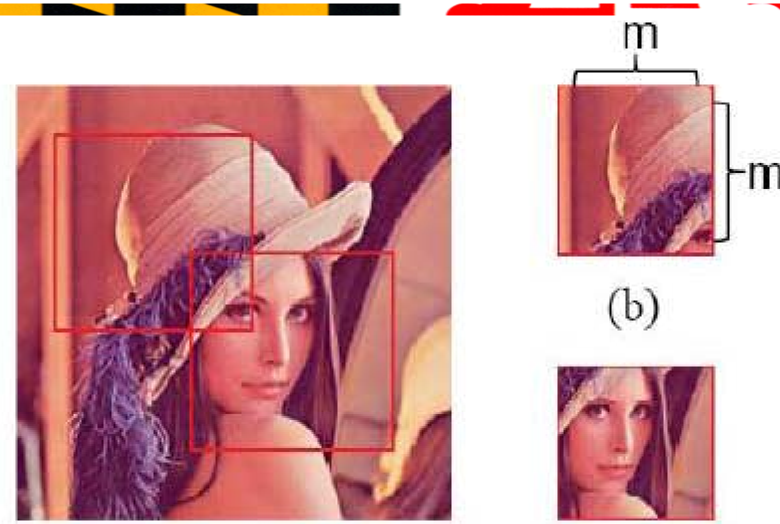
H is a *d*-by-*d* normalized Hadamard matrix:

D is a *d*-by-*d* diagonal matrix

FJLT-based Image Hashing

- Random sampling (secret key)
- *Sub(i)* is a vector with length $d = m^2$

• ***Original Feature Matrix*** *=*
{Sub(1), Sub(2), … , Sub(N)}, with d-by-N



(a) (b) (c)

• *Dimension Reduction using FJLT:*
Intermediate Hash *= FJLT (**Original Feature Matrix**), with k-by-N*

- *Random Weight Incorporation:* Generate Random Weight Matrix $W = \{w_1, w_2, \dots, w_N\}$

$$Hash = \{\langle IH_1, w_1\rangle, \langle IH_2, w_2\rangle, \dots, \langle IH_N, w_N\rangle\}, \text{ with } 1 \times N$$

$$X = \overbrace{\begin{bmatrix} S_{1,1} & \cdots & S_{1,N} \\ \vdots & \ddots & \vdots \\ S_{d,1} & \cdots & S_{d,N} \end{bmatrix}}^{N} \xrightarrow{d} X' = \overbrace{\begin{bmatrix} IH_{1,1} & \cdots & IH_{1,N} \\ \vdots & \ddots & \vdots \\ IH_{k,1} & \cdots & IH_{k,N} \end{bmatrix}}^{N} \xrightarrow{k} X'' = [FH_1, FH_2 \dots, FH_N]$$

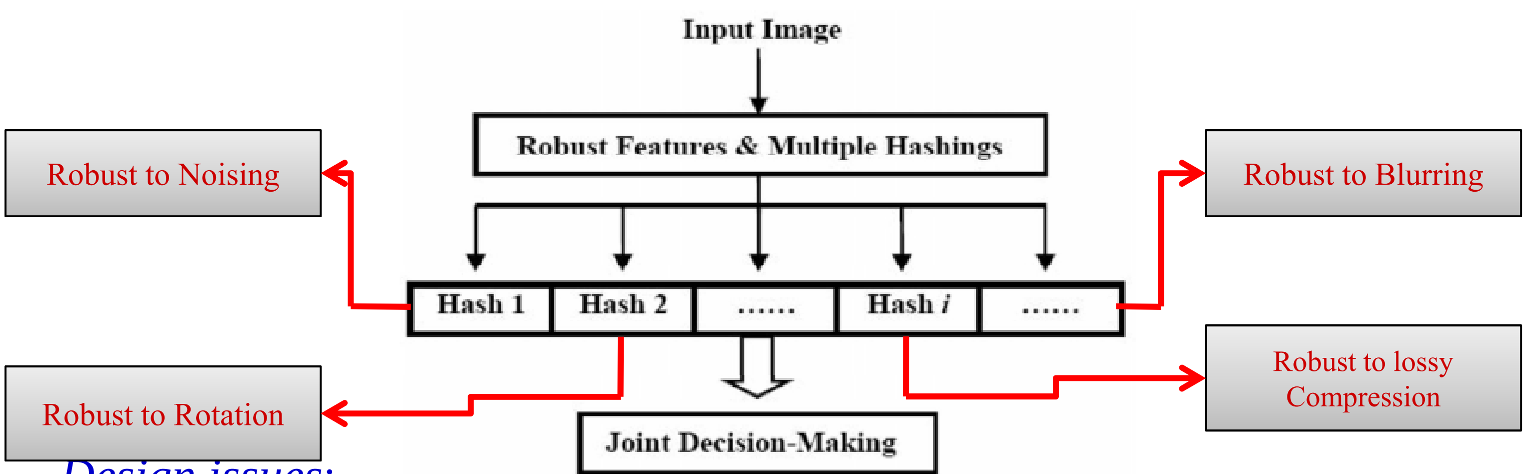
Proposed Image Hashing Algorithm

Database: 100 original images, each of them has 99 distorted copies=10000 images

Manipulations	Parameters setting	No.	NMF	FJLT
Additive Noise				
Gaussian Noise	Sigma: 0~0.2	10	49.7%	76.5%
Salt&Pepper Noise	Sigma: 0~0.2	10	75.9%	96.1%
Speckle Noise	Sigma: 0~0.2	10	50.2%	98.1
Blurring				
Gaussian Blurring	Filter size: 3~21, sigma=5	10	99.3%	100%
Circular Blurring	Radius: 1~10	10	99.4%	100%
Motion Blurring	Len: 5~15, theta: 0~90	9	99.7%	100%
Geometric Attacks				
Rotation	Degree= 5~45	9	19%	36.67%
Cropping	5%, 10%, 20%, 25%, 30%, 35%	6	15.83%	92.5%
Scaling	25%, 50%, 75%, 150%, 200%	5	99.8%	100%
JPEG Compression	Quality Factor=5~50	10	99.9%	100%
Gamma Correction	Gamma= (0.75~1.25)	10	5.4%	87.1%

Combining Hashing Algorithms

- ***Motivation***: *Combining* more than one image hashing algorithm to overcome their individual deficiencies
- ***Pros***: High identification accuracy under diverse distortions and manipulation; Multilayer security arising from the different hash generation processes
- ***Cons***: Increasing computational complexity



- *Design issues*:
  - Robust feature extraction methods for each image hashing algorithm, which are robust to specific types of attacks
  - Advanced joint decision making

Fourier-Mellin Transform (FMT)

Fourier-Mellin transform makes input image pattern invariant to geometric attacks including translation, rotation and scaling

Motivation of RI-FJLT Image Hashing: The input feature is rotation-invariant before FJLT hashing

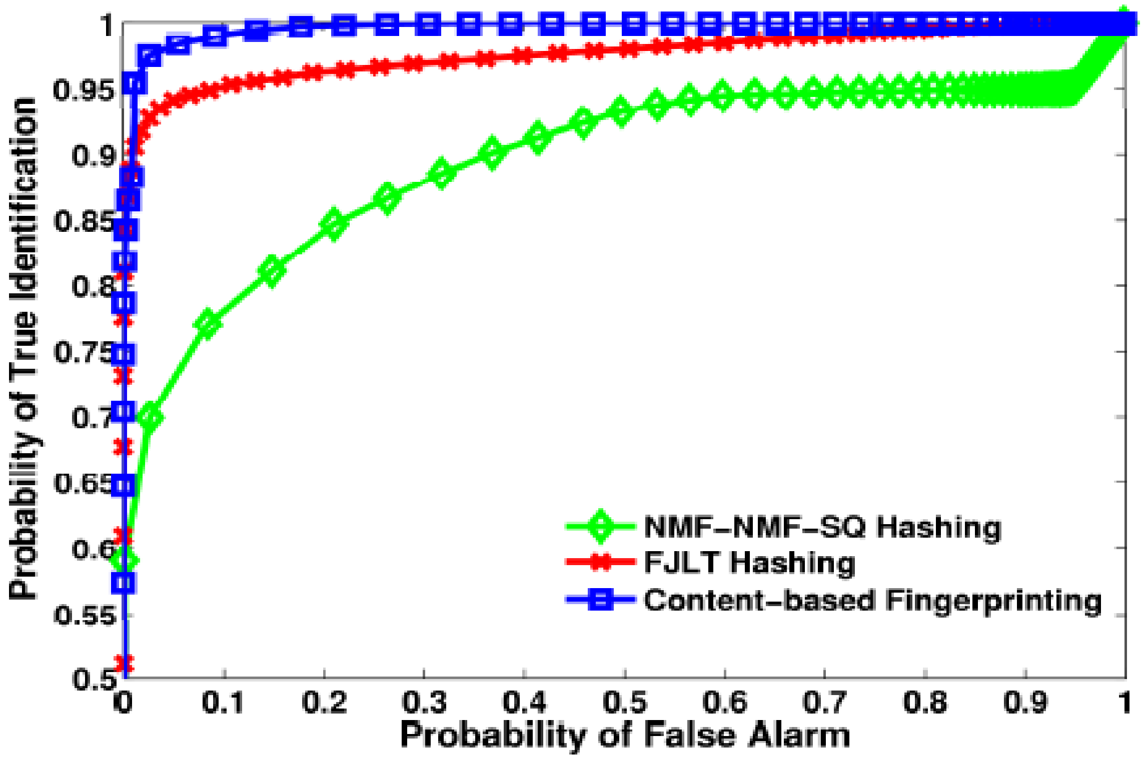
Content-based Fingerprinting using FJLT and RI-FJLT Image Hashing

Image Database: 100 original images, each of them has 99 distorted copies=10000 images

Manipulations	Parameters setting	No.	NMF	FJLT	FJLT+ RI_FJLT
Additive Noise					
Gaussian Noise	Sigma: 0~0.2	10	49.7%	76.5%	72.4%
Salt&Pepper Noise	Sigma: 0~0.2	10	75.9%	96.1%	95.9%
Speckle Noise	Sigma: 0~0.2	10	50.2%	98.1	98.2%
Blurring					
Gaussian Blurring	Filter size: 3~21, sigma=5	10	99.3%	100%	100%
Circular Blurring	Radius: 1~10	10	99.4%	100%	100%
Motion Blurring	Len: 5~15, theta: 0~90	9	99.7%	100%	100%
Geometric Attacks					
Rotation	Degree= 5~45	9	19%	36.67%	90.67%
Cropping	5%, 10%, 20%, 25%, 30%, 35%	6	15.83%	92.5%	90%
Scaling	25%, 50%, 75%, 150%, 200%	5	99.8%	100%	100%
JPEG Compression	Quality Factor=5~50	10	99.9%	100%	100%
Gamma Correction	Gamma= (0.75~1.25)	10	5.4%	87.1%	83.5%

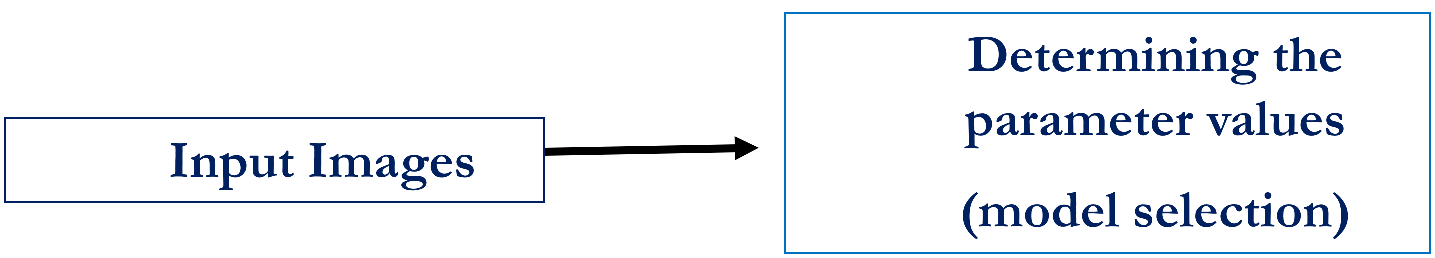
Content-based Fingerprinting using FJLT and RI-FJLT Image Hashing

Receiver Operating Characteristics (ROC) Analysis



Conclusion: Content-based fingerprinting achieves higher probability of true identification under the same false alarm rate.

Automation of Image Hashing Algorithms

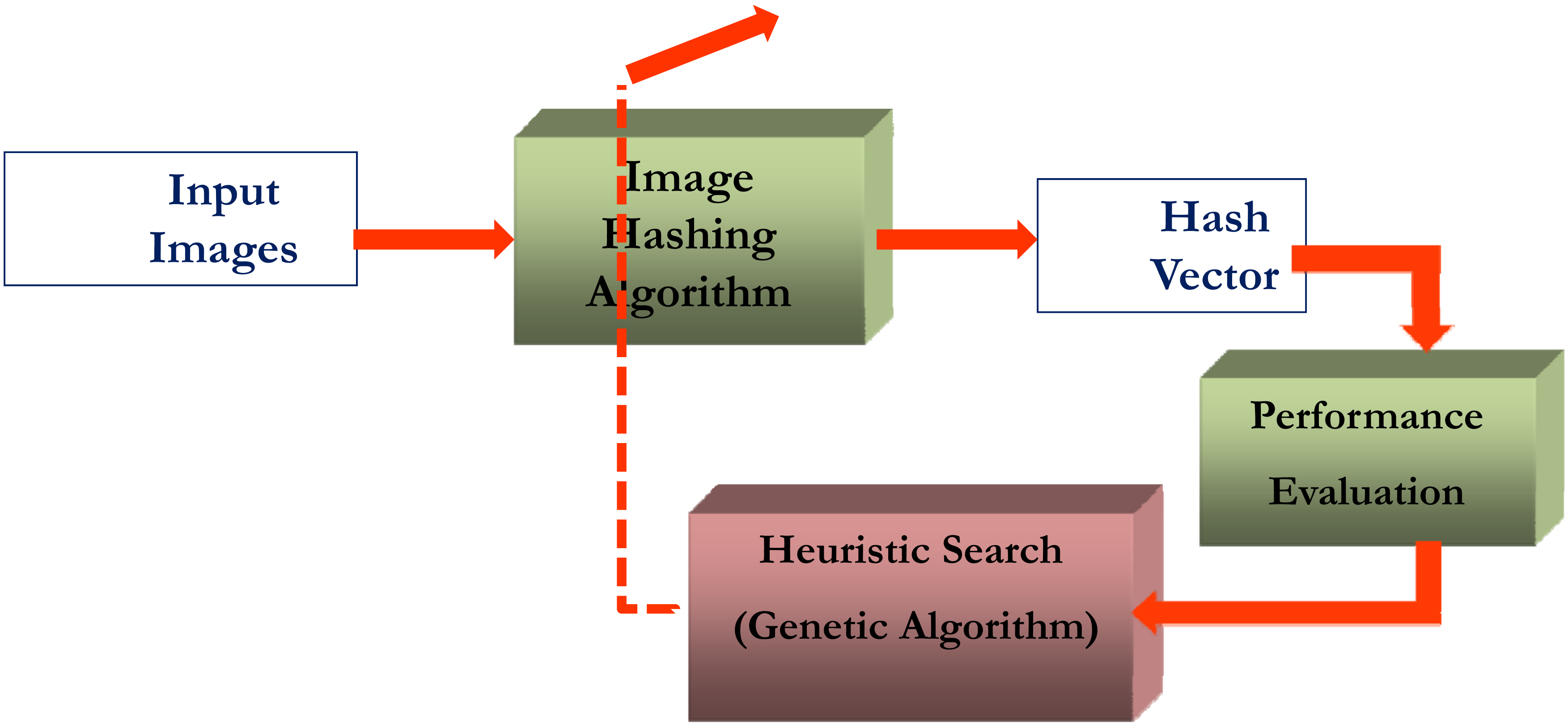


Automation of Image Hashing Algorithms

Manual Tuning →
- -Not optimal
- -Time consuming
- -Not suitable for large dimensions

Automatic Tuning →
- -Can lead to (sub)-optimal solution in a limited number of iterations
- -Time efficient
- -Suitable for large dimensions

Automation of Image Hashing Algorithms

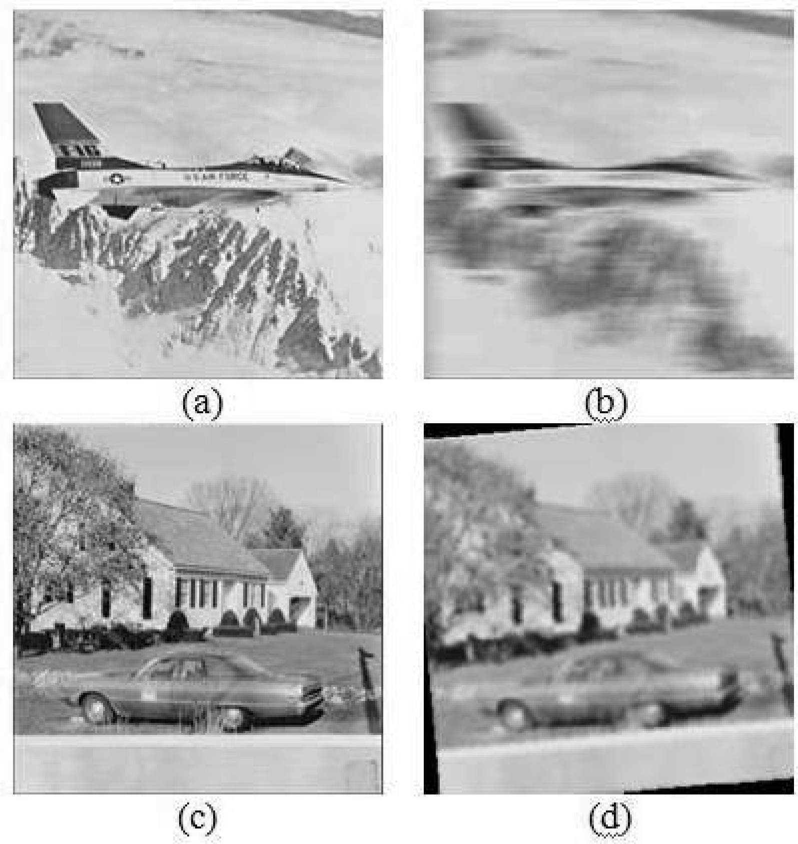


Results of Automating FJLT using GA

- Used a database of 50 standard images.

- Generated 20 attacked version of each image (composite attacks).

- Used half of the dataset for training the automatic parameter selection algorithm and the other half for testing the performance.



(a) (b) (c) (d)

Results of Automating FJLT using GA

Table 1. Comparison of optimal and manual results

Method	Parameter values	TPR (%)	FPR (%)	Kappa	Time (sec.)
Manual	[64 20 2.0 0.1 250 0.29]	89.6	4.3	0.83	2118.2
Optimal	[16 128 1.1 0.21 200 0.37]	95.6	6.3	0.89	42.0

* Results showed performance g improvement (measured by Kappa) as well as speed improvement compared to the original parameter setting (manual).

* Future work involves implementing the same procedure for other state-of-the-art image hashing algorithms and comparing the results.

Video Copy Detection

- Detect transformed copies of a video
- Represent the video with a fingerprint that is based on the content of the video
- Video content: Visual and Auditory
- Current approaches:
  - Image-based
    1. *uses every frame*
    2. *uses only key frames*
  - The whole video
- Problems
  - Using every frame: computationally not efficient
  - Using key frames: sensitive to noise

Video Detection-Our Approach

- Our approach: Make temporally informative representative images (TIRI) using weighted averaging of subsequent frames

$$o_{m,n} = \sum_{k=1}^{L} w_k \, l_{m,n,k}$$

- $l_{m,n,k}$ is the luminance value of the $(m, n)^{th}$ pixel of the k^{th} frame in a set of L frames.
  - $w_k = 1$ (simple averaging)
  - $w_k = k$ (linearly changing weights)
  - $w_k = 1 - e^{(k-\mu)^2/\sigma}$ (Gaussian weighting)
  - $w_k = \gamma^k$ (exponential weighting)

Averaged using Gaussian weighting

Averaged using Exponential weighting

Video Detection- Our Approach

How our video hashing algorithm works:

- Create the TIRI images for the video sequence
- Input the resulting TIRI images into the hashing algorithm
- We used a simple yet efficient hashing algorithm

  Low freq. DCT coefficients of the TIRI

Performance Evaluation

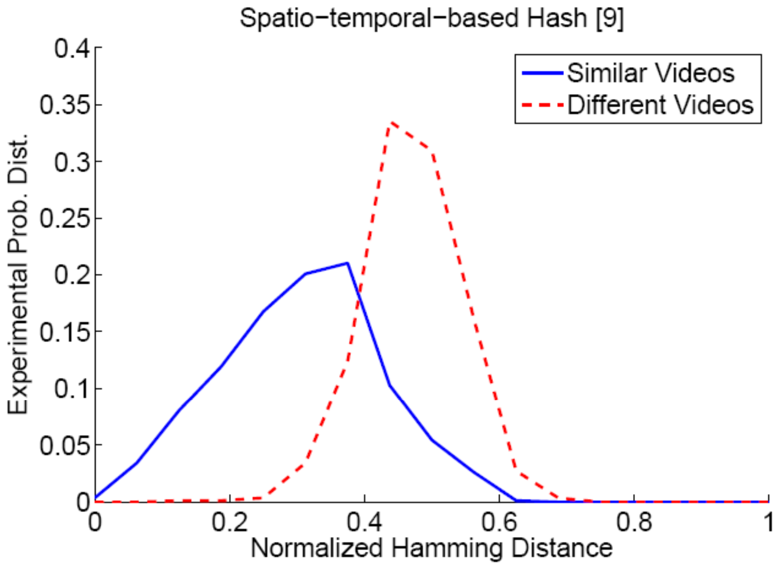
- Created 10 attacked versions of 14 videos (140 videos in total)

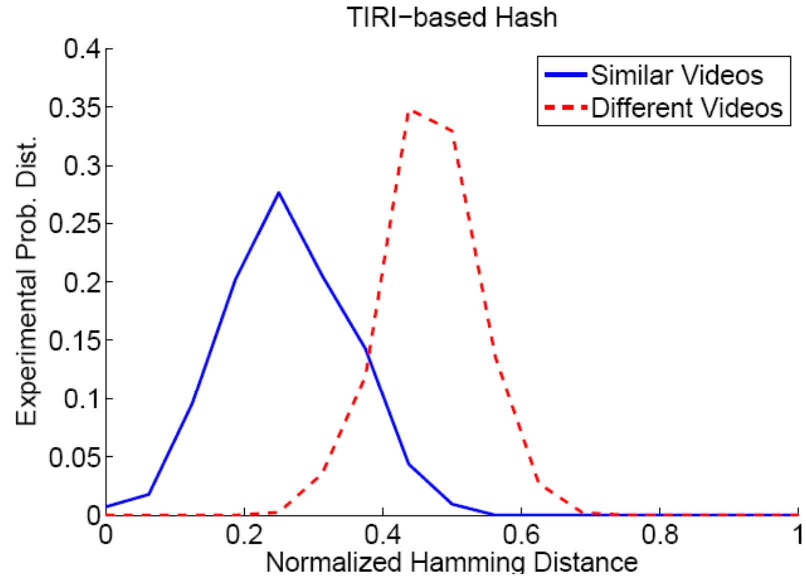
Attack	Effect	Min	Max
Noise (σ)	$l'_{m,n,k} = l_{m,n,k} + G(0,\sigma)$	0	100
Brightness (b)	$l'_{m,n,k} = l_{m,n,k} + b\,\mu_k$	-0.7	0.7
Contrast (c)	$l'_{m,n,k} = c\,(l_{m,n,k} - 127.5)$	0.5	2
Rotate (r)	Rotate the whole frame $r\,^\circ$	-5	5
Time shift (δ)	Video is shifted for δ seconds in time	-0.5	0.5
Spatial shift (sr, sd)	Shift the frame $sr\%$ to the right and $sd\%$ down	-4	4
Frame drop (fd)	$fd\%$ of the frames are randomly dropped	0	65

Performance Evaluation(sample attacks)

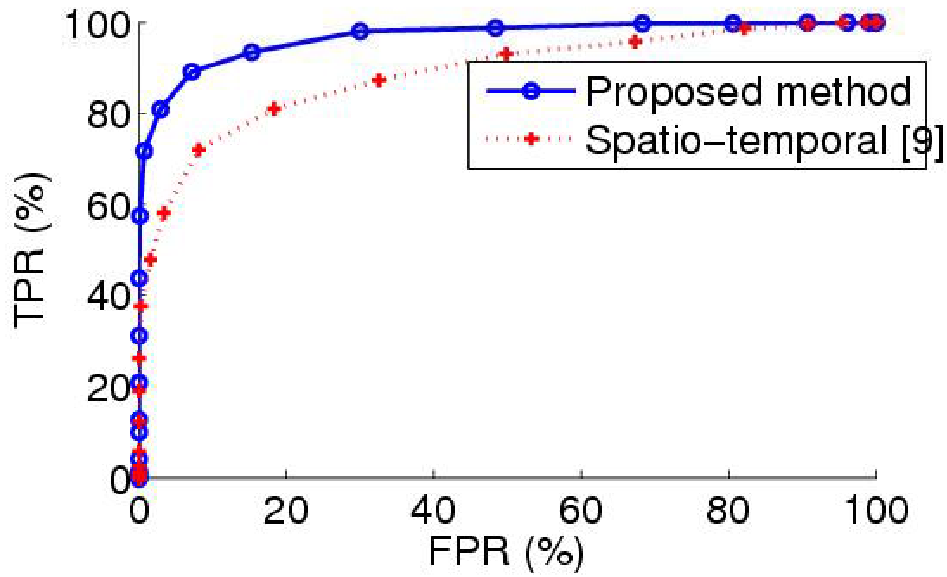
Performance Comparison

- Probability distribution of hash differences
  - *TIRI-based hash*
  - *Low freq. DCT coefficients of the whole video (Spatio-temporal-based hash)*





Performance Comparison



Performance Comparison

Attack	Spatio-temporal [2]		TIRI (%)	
	TPR(%)	FPR(%)	TPR(%)	FPR(%)
Noise	99.9	0.14	100	0
Brightness	99.8	0.2	100	0
Contrast	100	0	100	0
Rotation	100	0	98.2	0.4
Time shift	84.3	15.7	98.8	0.5
Spatial shift	100	0	96.8	1.8
Frame drop	97.1	2.9	99.2	0
Average	97.3	2.9	99.2	0.4

Performance Comparison (speed)

Method	No. of Frames in the Segment			
	4	8	16	32
DCT	30mSec	45mSec	80mSec	160mSec
TIRI	5mSec	7mSec	11mSec	18mSec
Speed ratio	6	6.4	7.3	8.9

Summary

- Dimension reductions techniques (e.g. FJLT) are promising for image hashing.
- Automating as well as combining image hashing algorithms can yield better identification performance.
- Creating a representative image from a video chunk can lead to a superior performance for video detection.

Behavior Dynamics in Multimedia Social Networks

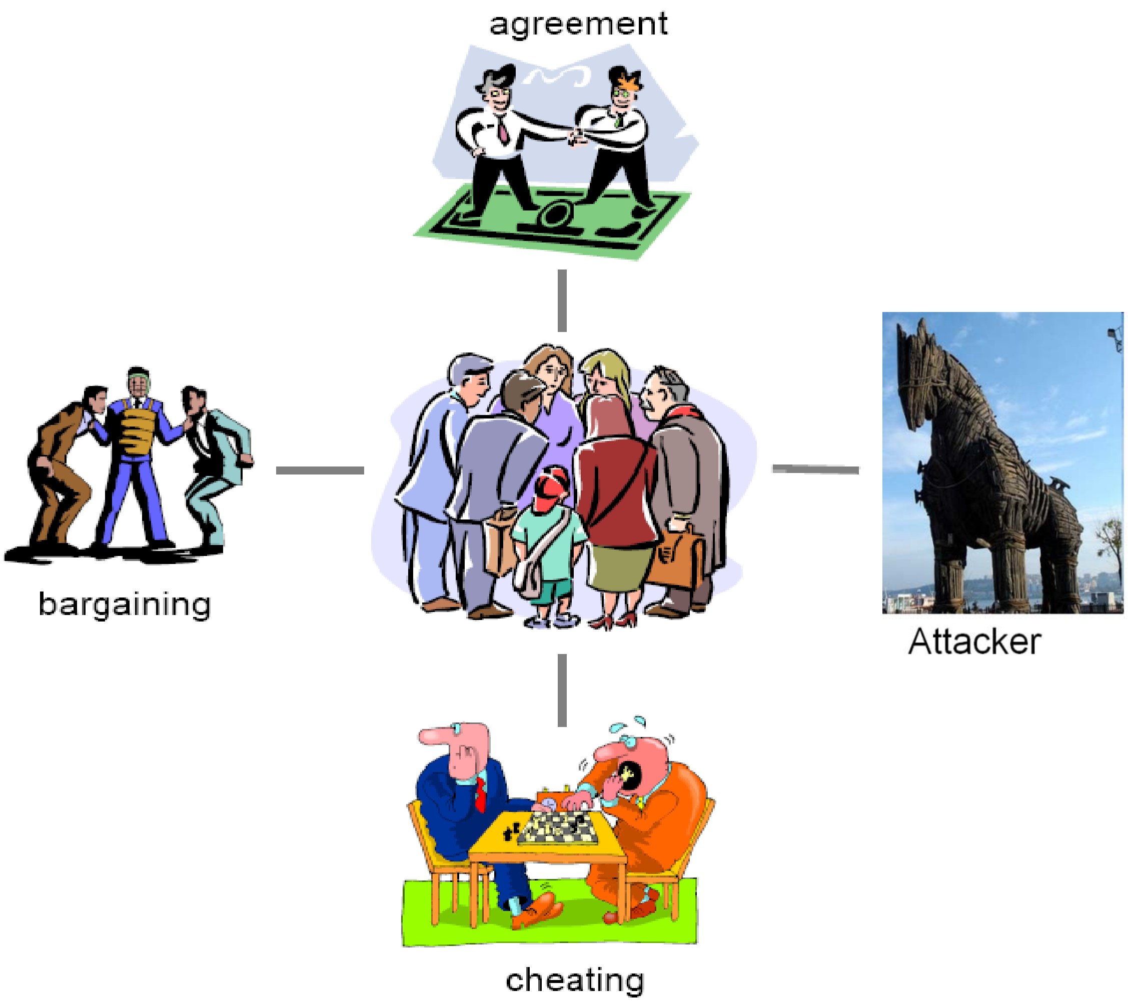
H. Vicky Zhao

ECE Dept., University of Alberta, Canada

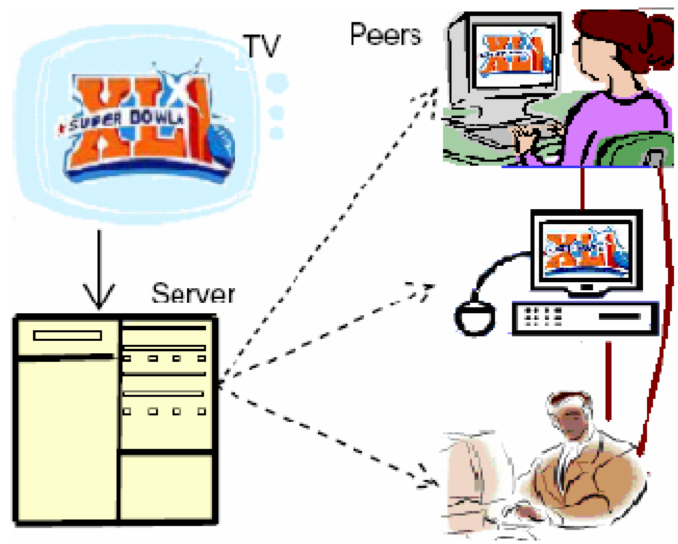
Acknowledgement: Ms. Sabrina Lin and Prof. K. J. Ray Liu from University of Maryland, Mr. Bo Hu from University of Alberta

Recall – Multimedia Social Networks

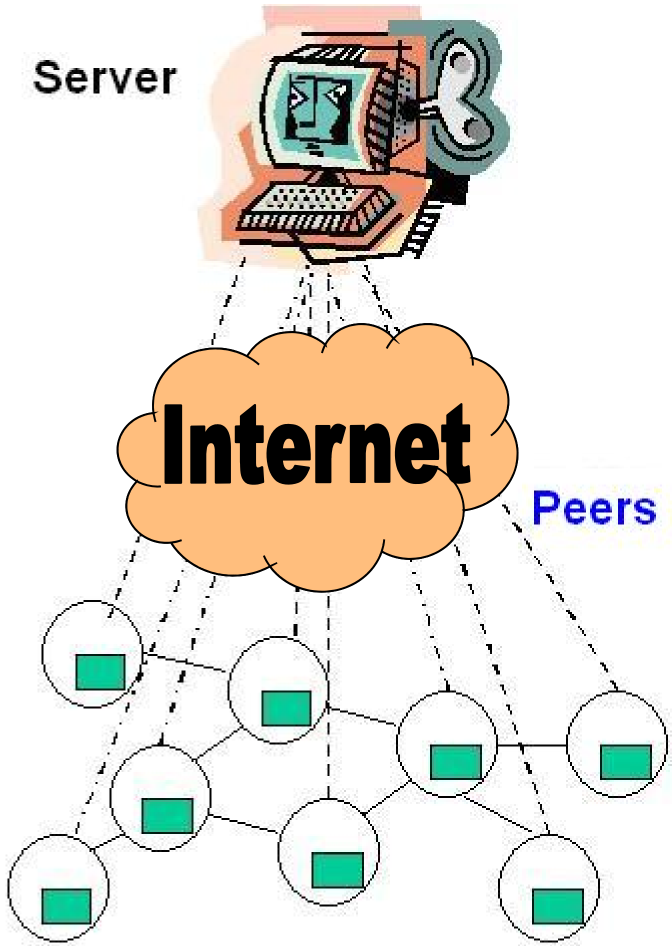
- Multimedia social networks: user interaction



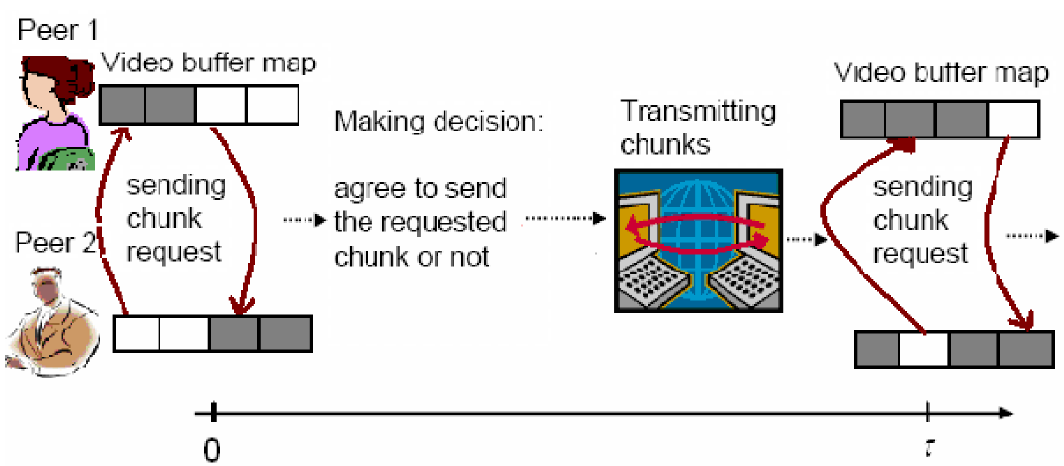
P2P live streaming



Mesh-Pull P2P Live Streaming



- Original server divides video into media chunks of *M* bits
- Every peer requests one chunk at the beginning of every round
- Each peer decides to answer or reject the chunk request



User Dynamics in P2P Live Streaming

- Selfish (rational) users:
  - Goal: receive a high-quality video and upload fewer chunks
  - Free riding
  - Might cheat if cheating can help increase their payoffs
- Malicious users (attackers)
  - Goal: maximize the damage to the system
  - Pollution attack: send unusable chunks
  - Hand wash (whitewashing)
- Our goal:
  - Stimulate cooperation and prevent cheating behavior
  - Minimize the damages caused by attackers

Cooperation Stimulation

- Two-player game model: request/upload at most 1chunk/round
- Utility definition: for each round,
  - User 1's utility:

Number of chunks that user 1 sends in this round

Number of chunks that user 2 sends in this round

User 1's cost (upload bandwith) of uploading a chunk to user 2

$$\pi_1(a_1, a_2) = (a_2 P_{21}) g_1 - a_1 \boxed{\frac{M}{W_1 \tau}}$$

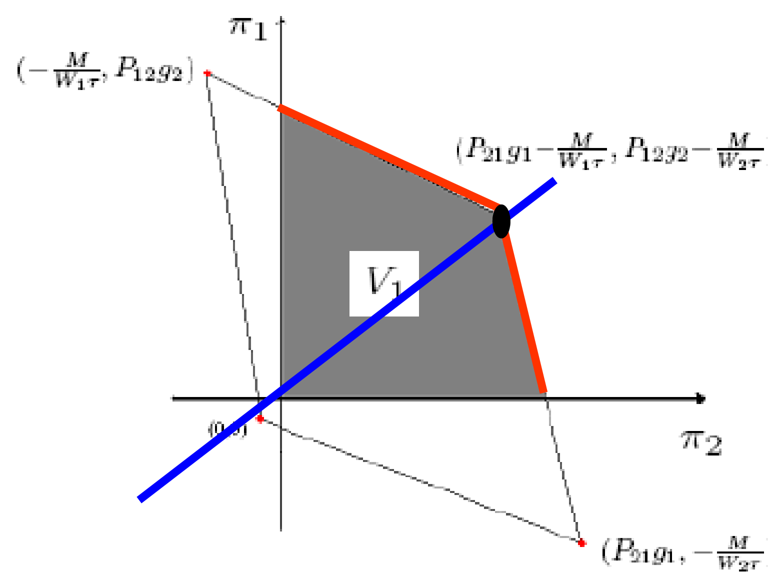
User 1's gain of receiving a chunk from user 2

  - User 2's utility: $\pi_2(a_1, a_2) = (a_1 P_{12}) g_2 - a_2 \dfrac{M}{W_2 \tau}$

Infinite-Time Model

- Infinite-time model : game termination time unknown
  - Strategy profile: $s=(s1=[a_1^1, a_1^2, a_1^3, \ldots], s2=[a_2^1, a_2^2, a_2^3, \ldots])$
  - Average utility: $U_i(s_1, s_2) = \lim_{T\to\infty} \sum_{t=1}^{T} \pi_i(s)/T$
  - $\geq$1 NE for every feasible and enforceable payoff profile

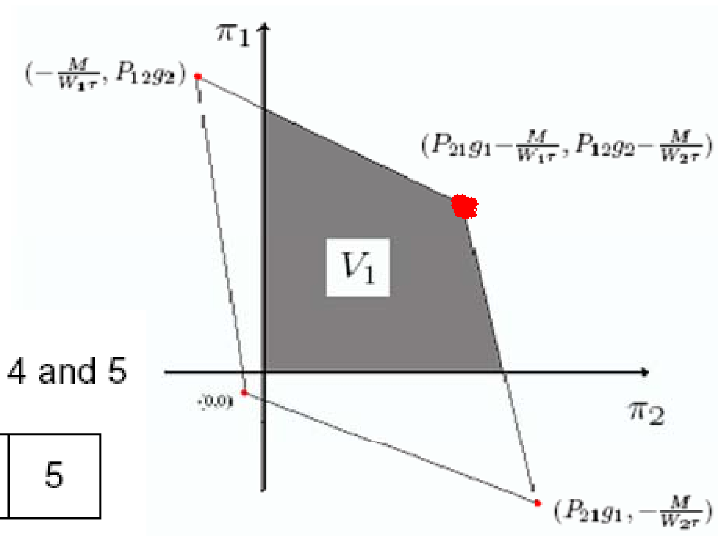
- Pareto-optimal set:
  - Each player cannot increase his/her utility without degrading others'
  - Rational users will always go
- Refinement by fairness criteria:
  - Absolute fairness
  - Proportional fairness



Cheat-Proof Strategies

$$\pi_1(a_1, a_2) = (a_2 P_{21}) g_1 - a_1 \frac{M}{W_1 \tau}$$

- Cheat on private information (g_i, W_i, P_{ji}):
  - Both players will report **false** private information to max. their own utilities under the constraint $P_{ji} g_i \geq M/(W_i \tau), W_i \geq W_{\min}$
  - Absolute and proportional fairness solutions become:
    - $x^*=(1,1)$ → *always cooperate*



- Cheat on buffer information

User i's buffer

1	2	3	4	5

Cheated buffer by hiding chunk 4 and 5

User i's buffer

1	2	3	4	5

  - Users send equal number of chunks to each other

- This solution is cheat-proof, Pareto-optimal Nash equilibrium

[1] W.S. Lin, H.V. Zhao, and K. J. R. Liu, "Incentive Cooperation Strategies for Peer-to-Peer Live Streaming Social Networks", , *IEEE Tran. on Multimedia*, vol. 11, no. 3, pp 396-412, April 2009

Pollution Attacks and Trust Model

- Pollution attack: upload useless chunks
  - Challenges: “intentional” vs “innocent” misbehavior
  - Attackers’ hand wash makes it much more challenging

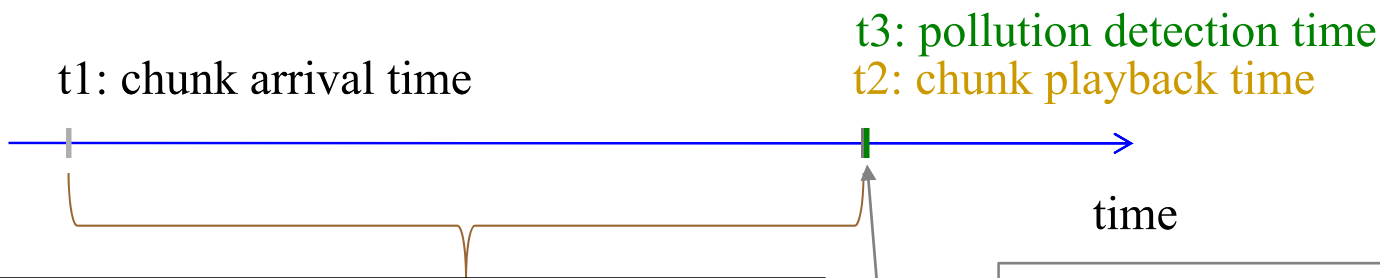
- Trust: the confidence that user i has on j to upload a clean chunk
  - User i identifies user j as malicious if $T_{i(j)}(t) < TH$
  - Collect the network opinion and identify attackers **early**

$$T_{i(j)}(t) = \beta \cdot \underbrace{DV_{i(j)}(t)}_{\text{user i's own experience}} + (1-\beta) \cdot \underbrace{IDV_{i(j)}(t)}_{\text{other users' opinion about j}}$$

Speed up the detection process

Delay in Polluted Chunk Detection

- In current P2P live streaming systems, a data chunk is not processed (decoded) until its playback time
  - a polluted data chunk **cannot** be detected until it is processed

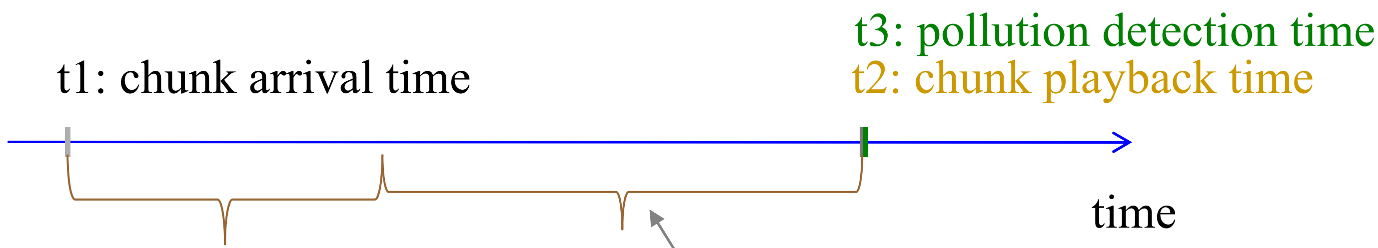


A selfish user may unintentionally forward polluted chunks to other users
→ Propagation of polluted chunks
→ Increase false alarm rates
→ **no cooperation among selfish users**

T3=t2: when a polluted chunk is detected, it's too late to ask for a clean version due to time constraint
→ Quality degradation

Early Detection of Polluted Chunks

- **Early** detection of polluted chunks: reduce the delay [2][3]
  - Extra overhead and increased complexity



• Faster detection of attackers
→ prevent them from uploading more polluted chunks
• A selfish user sends fewer polluted chunks
→ Reduce false alarm rates
→ **stimulate cooperation among selfish users even under attacks**

t3<t2: give the user a second chance to get a clean version
→ Quality improvement

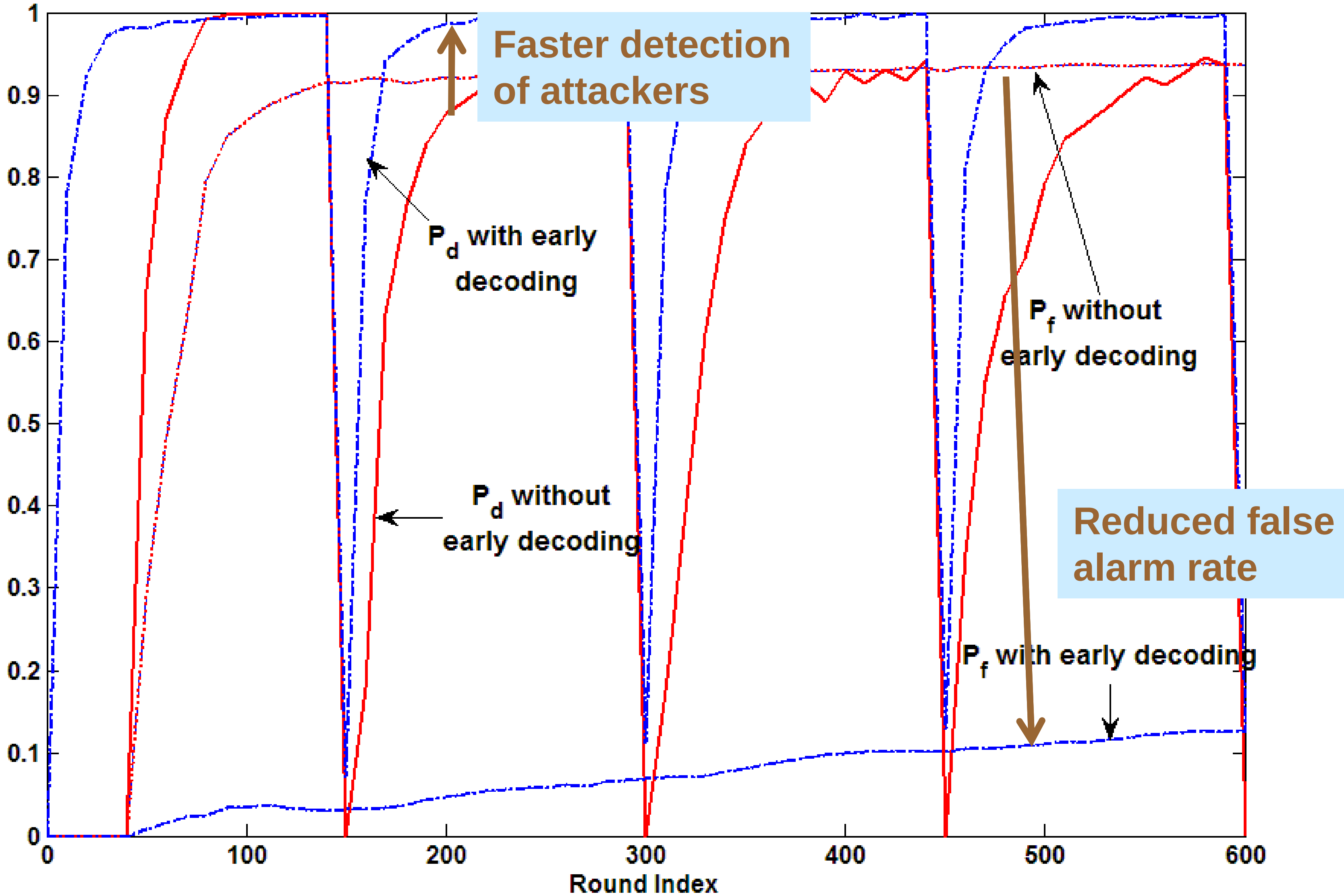
[2] P. Dhungel, X. Hei, K. W. Ross, and N.Saxena, "The pollution attack in P2P live video streaming: measurement results and defenses," *ACM SigComm Workshop on P2P Streaming and IP-TV*, pp. 323–328, Aug. 2007.
[3] Bo Hu, H. Vicky Zhao, "Pollution-resistant peer to peer live streaming using trust management", *to appear, ICIP 2009.*

Simulation Setup

- # of users: 144, # of malicious attackers:10
- Buffer length: 30 seconds' video
- Round duration: 1/9 second
- Video bit rate: 64Kbps

- Attackers:
  - Send polluted chunks whenever possible
  - Hand wash every 150 rounds
- Selfish users:
  - For data chunks in the first 20% of the buffer, a user can verify their authenticity immediately after their arrival

Simulation Results



Summary

- Behavior dynamics is an important issue in MM social networks
- User dynamics in P2P live streaming
  - A game-theoretic model for user behavior modeling
  - Cheat-proof cooperation stimulation strategies
  - Attack-resistant P2P system

Discussion

- Quality monitoring:
  - What partial information (e.g., hash, watermark) to use?
  - How to achieve adaptive and automated design?
- Content identification:
  - What feature descriptors to combine? And how to do the fusion?
  - How to combine both visual and audio features for video identification?
- Behavior dynamics:
  - Tradeoff between the robustness and the complexity
  - The impact of the social networks' structures on user interaction and behavior dynamics
  - …