


# Web a multimédia, PCA

doc. Ing. Josef Chaloupka, Ph.D.



# DVB – T2

- (od 25. října 2005) 2011 – přechod z analogového na digitální vysílání DVB – T, rozhlas - 2025
- Kodek H.265 (HEVC) – kódování MPEG4 - omezení z frekvencí 700 MHz – pro mobilní telefony
- Příjem v ČR – DVB – T2 + kodek HEVC 
- Zakódování komerčních kanálů – potřeba dalších technických zařízení
- Anténní příjem – DVB –T2 X Internetová televize, satelit
- Změna kvality obrazu – většinou jen nižší rozlišení
- 4 větší komprese při „stejně“ kvalitě dat jako DVB – T (MPEG2)
- 2021 vypnutí DVB – T, regionální sítě (červen 2019), celoplošné (listopad 2019), Praha (leden 2020)
- Vysílání se přesune na stejnou frekvenci – anténa může přijímat z více vysílačů najednou – očekávaná lepší kvalita přijímaného signálu

# DVB – T2 - přínosy

- Novější kódování (MPEG4) a kodek H.265 (HEVC)
- Větší hustota programů v jednom kanálu při cca stejné kvalitě vysílání jako DVB–T
- Kapacita datové toku cca o 30% vyšší
- „Možnost“ vysílání ve vyšším rozlišení
- Různý typ dat v jednom kanále – HDTV, mobilní příjem ...
- Odolnost vůči ztrátě dat – cca o 7 dB lepší
- Větší odstup nosných kmitočtů – eliminace Dopplerova jevu
- Možnost placených kanálů
- Itálie, VB – zvýšení přenosové rychlosti z 19,91 na 33,3 Mb/s (+67%)



# HTML 5

- **HTML5:** <https://html5.org/> <http://www.html5.cz/>
- V počátcích 90. let s příchodem internetu
- HTML 5 – finální specifikace 28.10.2014
- HTML 4 -> 1997
- Přehrávání multimédií, podpora (off-line) aplikací bez přístupu na síť
- Nové nebo modifikované tagy
- Perzistentní (trvalé) úložiště formou asociativního pole
- Relační databáze s podporou transakcí
- Zpětná kompatibilita?

# HTML 5 – Struktura stránky

- **nové (modifikované) elementy:**
- **!DOCTYPE** – nepárová značka – zkrácení: od HTML 5 není potřeba uvádět verzi:  
`<!DOCTYPE html>`

## STRUKTURA STRÁNKY

- **article** – samostatná a nezávislá část stránky - komentář, článek...; možný obsah:  
header, footer, další article...  

```
<article>  
    <header>  
        <h3>Můj první článek</h3>  
        <time datetime="2014-04-01">01. dubna 2014</time>  
    </header>  
    <p>V mém prvním článku bych...</p>  
</article>
```
- **aside** – obsah vztahující se k hlavnímu obsahu stránky; využití: boční panel na stránce; nevnořovat do elementu **address**
- **section** – obsah stránky (KAPITOLY), který při vyjmutí poruší logiku stránky x
- **article** – nezávislý článek
- **main** – hlavní část stránky

# HTML 5 - Struktura stránky

- **footer** – zápatí stránky: jméno autora, odkaz, informace o autorských právech...
- **header** – hlavička sekce
- **cite** – i v předchozích verzích; upřesnění – titulek citovaného díla
- **details** – podrobnosti o (části) dokumentu – lze odtud získat další informace o dokumentu, nevnořovat do elementu **a** a **button**; atribut *open* – viditelnost obsahu
- **header** – hlavička sekce/stránky – vložení nadpisu a úvodního obsahu, navigační pomůcky, logo...; nevnořovat do elementu **address, footer**
- **hgroup** – seskupování nadpisů; obsah – nadpisy h1-h6

```
<article>
  <hgroup>
    <h1>Nová auta na trhu</h1>
    <h2>Škoda Octavia RS 2.0 TDI</h2>
  </hgroup>
  <p>Škoda Octavia...</p>
</article>
```
- **hr** – nový sémantický význam – tematické rozdělení textu na více částí

# HTML 5 – Struktura stránky

- **meta** – nově přibyl atribut *charset*; ubyl atribut *http-equiv=content-language*

`<meta charset="UTF-8">`

- **nav** – hlavní navigace: odkazy na hlavní části stránky nebo stránky s podobným obsahem; nevnořovat do elementu **address**
- **summary** – titulek – využití v elementu details

`<details>`

`<summary> Copyright 2010-2014.</summary>`

`<p>by ColorGroup. All Rights Reserved. </p>`

`<p>All content and graphics on this web site are the property of the company  
ColorGroup</p>`

`</details>`

- **time** – nastavení data, času; atributy: *datetime*, *pubdate* (publikační datum, čas)

`<time datetime="2014-04-03"> 3. dubna 2014 </time>`

`<time datetime="2014-04-03T13:00" pubdate> 3. dubna 2014 ve 13:00</time>`

# HTML 5 – Struktura stránky

- **mark** – zvýraznění/označení textu; výchozí nastavení text (červená barva), pozadí (žlutá barva)
- **menu** – seznam ovládacích prvků pro formuláře; atributy: *type* (toolbar – panelová nabídka; context – kontextová nabídka); *label* (popis nabídky) nevnořovat do elementu **a, button**

```
<strong>Konfigurace motoru Škoda Octavia:</strong>
  <menu>
    <br><input type="checkbox">2.0 TSI (162 kW)
    <br><input type="checkbox">2.0 TDI (135 kW)
  </menu>
```



# HTML 5 – Struktura stránky

● **input** – vstupní pole; různé typy obsahu; atributy *type*, *value*, *disabled*, *readonly*, hodnoty *type*: text, password, checkbox, radio, button, submit, reset, file, hidden, image, datetime, datetime-local, data, month, time, week, number, range, email, url, search, tel, color

Jméno:

Příjmení:

☒ Muž  
☐ Žena  
☐ Jiné

```
<form action="nejaka_akce.php">  
  Jméno:<br>  
  <input type="text" name="jmeno"><br><br>  
  Příjmení:<br>  
  <input type="text" name="prijmeni"><br><br>  
  <input type="radio" name="gender" value="muz" checked> Muž<br>  
  <input type="radio" name="gender" value="zena"> Žena<br>  
  <input type="radio" name="gender" value="jine"> Jiné<br><br>  
  <input type="submit" value="Odeslat">  
</form>
```

# HTML 5 - Multimédia

- **canvas** – kontejner pro grafiku (vektorovou) + vkládání obrázků, plátno pro dynamické vykreslování -> grafy reagující na ovládání uživatele
- **figure, figcaption** – figcaption potomek figure; samostatná část stránky, kde je obrázek, graf, video, část kódu; - doplňující text k obrázku příp. zdrojovému kódu

```
<figure>  
      
    <figcaption>Nový model Škoda Octavia RS 2.0 TDI.</figcaption>  
</figure>
```

# Obrázky (fotky) na www

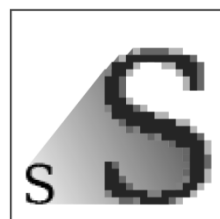
- **Formáty obrazu**

- Rastrová grafika (GIF (256 barev, průhlednost, animace), JPEG, PNG)
- Vektorová grafika SVG >>> Scalable Vector Graphics (škálovatelná vektorová grafika) >>> 2D vektorová grafika popsána pomocí XML
- 3 druhy grafických objektů >>>

- 1) vektorové tvary (obdélník, kružnice, elipsa, úsečka, lomená čára, mnohoúhelník a křivka)

- 2) rastrové obrazy

- 3) textové objekty



BITMAP  
.jpeg .gif .png



OUTLINE  
.svg

- Různé seskupování objektů, formátovány dle atributů, různé umístění v prostoru, geometrické transformace, animace, ořezávání objektů, filtrace obrazu >>> nemusí být podporováno ve všech prohlížečích

# Obrázky (fotky) na www

- Vložení obrázku (rastrová grafika) >>> element **IMG**

Atribut	Popis	Hodnoty
height	výška obrázku	px, %
width	šířka obrázku	px, %
src	URL obrázku	URL
align	zarovnání obrázku	left right top middle ...
alt	popis obrázků (nevidoucí osoby)	text
border	rámeček obrázku	px
vspace	velikost odsazení obrázku shora a zdola	px
hspace	velikost odsazení obrázku zleva a zprava	px
usemap	použití mapy	#jmeno_mapy
style	Inline CSS styl	CSS vlastnosti
id	Identifikace pomocí CSS id	název id
class	Identifikace pomocí CSS třídy	název třídy

- Příklad:
- ``
- Velikost obrázku (px) x zobrazující velikost
- Další využití obrázku na www: 1) odkaz, 2) animace, 3) pozadí, 4) klikací mapa...

# Obrázky (fotky) na www

- Element **SVG** (HTML5) :

```
<svg id="svgelem" height="200" xmlns="http://www.w3.org/2000/svg">
```

```
  <circle id="redcircle" cx="50" cy="50" r="50" fill="red" />
```

```
  <rect id="redrect" width="300" height="100" fill="red" />
```

```
  <line x1="0" y1="0" x2="200" y2="100" style="stroke:red;stroke-width:2"/>
```

```
  <ellipse cx="100" cy="50" rx="100" ry="50" fill="red" />
```

```
  <polygon points="20,10 300,20, 170,50" fill="red" />
```

```
  <polyline points="0,0 0,20 20,20 20,40 40,40 40,60" fill="green" />
```

```
  <image x="0" y="0" width="60" height="45" xlink:href="fotka.jpg"/>
```

```
  <text x="20" y="40" fill="white" font-family="Verdana">nějaký text</text>
```

```
</svg>
```



# Obrázky (fotky) na www

- Element **SVG** (HTML5) :

```
<!DOCTYPE html>
```

```
<html>
```

```
  <body>
```

```
    <svg width="300" height="200">
```

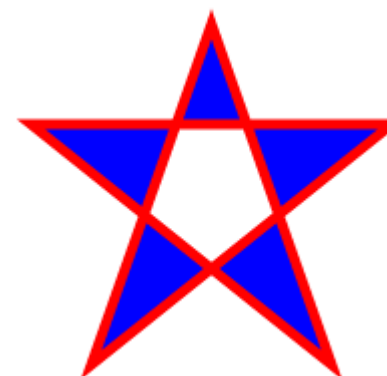
```
      <polygon points="100,10 40,180 190,60 10,60 160,180"
```

```
      style="fill:blue;stroke:red;stroke-width:5;fill-rule:evenodd;" />
```

```
    </svg>
```

```
  </body>
```

```
</html>
```



# Zvuk a video na www

- Zvukové formáty pro www: WAV, AU, MP3, RealAudio, ATRAC, WMA, Ogg, MIDI...
- 1 minuta záznamu (Audio CD kvalita) WAV (10MB) > MP3 (1MB) > MIDI (10kB)
- Video formáty pro www: AVI, MPEG, Windows Media, Real MediaFlash Video, QuickTime...
- Video >>> velké datové toky >>> rychlost připojení >>> komprese videa x kvalita videa
- Spouštění videa přes Flash
- **Autorská práva k hudbě a videu!**

# Vkládání multimediálních dat (video, zvuk)

- Využití tagu **embed** (starší), **object**

- embed atributy:

**src** zdroj dat (URL)  
**hidden** skrytí objektu  
**name** jméno pro spolupráci s jinými objekty  
další atributy >>> dle MM dat

- object atributy:

**data** zdroj dat (URL)  
**type** typ objektu  
**name** jméno pro spolupráci s jinými objekty  
**codebase** možné odkazy (URL)  
**clsid** přehrávací program  
**code** adresa programu v Javě  
**standby** hláška při načítání

- Kombinace obou tagů

```
<object data=„pokus.avi" type="application/avi" width="100" height ="100">  
  <embed src="earth.mpeg" type="application/avi" width ="100" height ="100">  
  </embed>  
</object>
```



# Vkládání multimediálních dat (video, zvuk) HTML 5

- Tag **video** a **audio** (HTML5), definice API pro spouštění...

```
<video src="URL "  
      width="šířka video okna"  
      height="výška videa okna">  
    ... další atributy  
</video>
```

```
<audio src="URL ">  
    ... další atributy  
</audio>
```

- **audio** – problematické přehrávání některých kodeků (Ogg Vorbis v IE nepodporován); atributy:

*autoplay* (okamžité přehrání)

*controls* (ovládací prvky)

*loop* (opakované přehrávání)

*preload* (načtení zvukové stopy před přehráváním) = bez hodnot

*src* -> odkaz na audio soubor

# Vkládání multimediálních dat (video, zvuk) HTML 5

## ● **video** – atributy:

*audio* -> možnost muted

*autoplay*

*controls*

*height* -> výška video přehrávače (v pixelech)

*loop*

*poster* -> odkaz na obrázek reprezentujícího

*videopreload*

*src* -> odkaz na video soubor

*width* -> šířka video přehrávače (v pixelech)

*type* -> typ (kodek) video souboru

```
<video width=„640“ height=„480“ controls autoplay poster=„skoda.jpg">
```

```
  <source src=„skoda.mp4“ type="video/mp4">
```

Nelze přehrát video ve vašem prohlížeči.

<br>možnost <a href=„skoda.mp4">stažení</a>

```
</video>
```

# Vkládání multimediálních dat (video, zvuk)

- **source** – alternativní verze, prohlížeč si vybere, využití několika (multimediálních) zdrojů dat; vnoření do elementu **audio** a **video**, prohlížeč si vybere formát, který „umí“ přehrát

*media* -> typ zařízení (SW), pro které jsou data optimalizovaná

*src* -> odkaz na datový soubor

*type* -> typ (kodek) datového souboru -> MME\_type

```
<video width=„640“ height=„480“ controls autoplay poster=„skoda.jpg">
```

```
  <source src=„skoda.mp4“ type="video/mp4">
```

```
  <source src=„skoda.ogg" type="video/ogg">
```

```
  <source src=„skoda.webm" type="video/webm">
```

Nelze přehrát video ve vašem prohlížeči.

```
  <br>možnost <a href=„skoda.mp4">stažení</a>
```

```
</video>
```

# Vkládání multimediálních dat (video, zvuk)

- Odkaz na soubor: `<a href=„hudba.mp3">hudba</a>`

- Youtube

```
<object data="https://www.youtube.com/embed/0xcyVBPoQbs"
      type="application/x-shockwave-flash" height="854" width="480">
  <param name="allowfullscreen" value="true" />
  <param name="allowscriptaccess" value="always" />
  <param name="src" value="https://www.youtube.com/embed/0xcyVBPoQbs" />
</object>
```

**IFRAME** -> plocha pro vložení jiné www stránky

```
<iframe width="854" height="480" src="https://www.youtube.com/embed/0xcyVBPoQbs"
frameborder="0" allowfullscreen>
</iframe>
```

# Streamování multimédií

- Streaming formáty: RealMedia, Windows Media, Flash Video, QuickTime...
- Kontinuální přenos audio(vizuálních) dat mezi serverem a PC uživatele >>> po Internetu (webcasting), přehrávají se průběžně stahovaná data (využití bufferu)
- Webcasting (internetová televize nebo rádio) >>> v reálném čase, Video on demand (Youtube)
- Datové toky >>> video 320×240 >>> 256–512 kb/s, >>> zvuk 16-256 kb/s
- Živé vysílání >>> 1) digitalizace signálu, 2) kódování signálu (v reálném čase) pro několik přenosových rychlostí, 3) streaming server posílá data >>> přenosová rychlost (datový tok) dle požadavku uživatele
- multicasting >>> posílání IP paketů z jednoho zdroje skupině více koncových stanic, nejsou posílány pakety ke každému zdroji zvlášť, ale je poslán jen jeden paket, jeden datový tok, zmenšení zátěže vysílajícího uzlu, ne každý paket musí dojít

# Synchronizovaná multimédia

- Klipy složené z různých multimediálních dat >>> zvuk, obraz, video, text
- Synchronizace multimédií na základě času, nebo je řízena dle interakce s uživatelem
- Nastavení umístění objektů na www + čas ke spuštění
- Menší datový tok u obrázků spojených se zvukem než u videa
- **FLASH**
- Od firmy Adobe
- **Silverlight**
- Od firmy Microsoft

# Synchronizovaná multimédia – Flash

- **FLASH**
- Grafický vektorový program – malá velikost výsledných souborů
- Interaktivní animace, prezentace, hry, video, flashové bannery
- Skriptovací jazyk ActionScript – vytvoření \*.swf souboru
- Vložení na www:

```
<object  
codebase="http://fpdownload.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=  
8,0,0,0" width=„500" height=„600" id=„prvni_pokus" align="middle">  
<param name="allowScriptAccess" value="sameDomain" />  
<param name="movie" value=„první_pokus.swf" />  
<param name="quality" value="high" />  
<param name="bgcolor" value="#00ff00" />  
<embed src="rozmazani.swf" quality="high" bgcolor="#00ff00" width="500" height=„600"  
name=„první_pokus" align="middle" allowScriptAccess="sameDomain" type="application/x-  
shockwave-flash" pluginspage="http://www.macromedia.com/go/getflashplayer" />  
</object>
```

# Synchronizovaná multimédia – Flash

```
package
{
    import flash.display.Sprite;
    import flash.events.Event;
    import flash.text.TextField;
    public class Main extends Sprite
    {
        private var text_1:TextField = new TextField();

        public function Main():void
        {
            if (stage) init();
            else addEventListener(Event.ADDED_TO_STAGE, init);
        }

        private function init(e:Event = null):void
        {
            removeEventListener(Event.ADDED_TO_STAGE, init);
            text_1.text = "Hello World!";
            text_1.x = 10;
            text_1.y = 5;
            text_1.textColor = 0x00FF00;
            addChild(text_1);
        }
    }
}
```



# Synchronizovaná multimédia – Flash

## ActionScript

-Přehrávání -> FlashPlayer; přípona \*.swf; podpora grafický a vývojový editor

```
<embed src=„skoda.swf" quality="high" width=„640" height=„480"
type="application/x-shockwave-flash"
pluginspage="http://www.macromedia.com/go/getflashplayer" />
```

-vykonávání událostí\_

```
-on(event){
    ...
}
```

press	-> kliknutí myši
release	-> uvolnění tlačítka myši
releaseOutside	-> uvolnění tlačítka myši mimo
rollOut	-> vyjetí kurzorem ven mimo oblast
rollOver	-> najetí kurzorem na oblast
keyPress(kód klávesy)	-> stisknutí tlačítka

# Synchronizovaná multimédia – Flash

## ActionScript

-MovieClip – události:

-onClipEvent(**ClipEvent**) {

...

}

load	-> po načtení MovieClipu
enterFrame	-> pro každý snímek (25 x za sekundu)
mouseMove	-> při pohybu kurzoru myši
mouseDown	-> kliknutí myši (kdekoli v dokumentu)
mouseUp	-> uvolnění tlačítka myši
keyDown	-> stisknutí klávesy
keyUp	-> uvolnění klávesy

- **trace** -> tisk textu do ouput okna trace("Hello world!");

- **stop()** -> ukončení přehrávání

- **gotoAndPlay()**, **gotoAndStop()** - přechod na snímek + akce; číslo snímku nebo symbolická proměná – definována v Properties

# Synchronizovaná multimédia – Flash

## ActionScript

**-proměnné** -> číslo; řetězec; boolean; null; undefined

```
-a = 20;  
-MyText = ' Flash je " dokonalý ' ' ;  
-MyText = " Flash je \" dokonalý \" \" ;  
-YesNo = true; "  
-ret = null;  
-\\r přidání řádku; \\t tabulátor
```

**-aritmetické operace** -> +; -; \*; /; =; %(modulo); +=; -=; \*=; /=; ++; --

```
-a = t + 20;  
-ret1 = "Flash je “;  
-ret2 = " \" dokonalý \" \" ;  
-MyText = ret1 + ret2;
```

# Synchronizovaná multimédia – Flash

## ActionScript

- **podmínka** -> >; <; >=; <=; ==; != (&&; ||; !výraz)

```
if (podmínka_1) {  
    ...  
} else if (podmínka_2) {  
    ...  
} else {  
    ...  
}
```

- **while, do...while** cyklus

```
while(výraz){  
    ...  
}  
do {  
    ...  
} while (výraz);
```

# Synchronizovaná multimédia – Flash

## ActionScript

### -for cyklus

```
for(i=0;i<10;i++){  
    ...  
}
```

### -pole

```
den = ["pondělí", "úterý", "středa", "čtvrtek", "pátek", "sobota", "neděle"];  
trace(den[2]); //středa
```

```
MyArr = [ [7,3] , [5,4] ];  
trace(MyArr[0][0]); //7  
trace(MyArr[1][0]); //5
```

```
for(a in den){ //procházení od konce  
    trace(den[a] + "\r");  
}
```

# Synchronizovaná multimédia – Flash

## ActionScript

### -funkce

```
function soucet(a, b) {  
    return a+b;  
}
```

```
c = soucet (10, 21);
```

**-adresace a objekty** -> objekty jsou symboly; this -> reference na aktuální instanci symbolu; \_parent -> reference na rodiče; with -> několikanásobný přístup k jednomu objektu

```
MyObject = new Object();  
MyObject.cislo = 10;  
trace(MyObject.cislo);  
MyObject.pozdrav = function(){  
    trace("Ahoj!");  
}
```

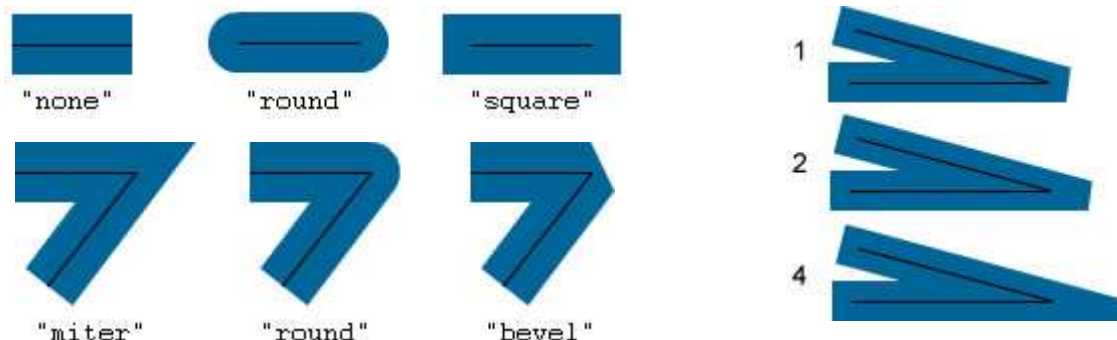
```
with(_root.myGame){  
    MyText = "nastav";  
    Position = 253;  
}
```

# Synchronizovaná multimédia – Flash

## ActionScript

- **Grafika**
- **1) čára**
- `lineStyle(šířka, RGB barva, průhlednost, zvětšení, konec čáry, typ spojení, ostrost spojení)`

- konec čáry
- typ spojení
- ostrost spojení



```
this.createEmptyMovieClip("obrazek", this.getNextHighestDepth());  
obrazek.lineStyle(2, 0x00FF00, 50, true, „none", "round", "miter", 2);  
obrazek.moveTo(500, 500);  
obrazek.lineTo(400, 400);  
obrazek.lineTo(300, 300);
```

# Synchronizovaná multimédia – Flash

## ActionScript

- **Grafika**
- **2) polygon**

```
var triangleHeight:uint = 100;  
// red triangle, starting at point 0, 0  
obrazek.beginFill(0xFF0000);  
obrazek.moveTo(triangleHeight / 2, 0);  
obrazek.lineTo(triangleHeight, triangleHeight);  
obrazek.lineTo(0, triangleHeight);  
obrazek.lineTo(triangleHeight / 2, 0);
```

- **3) křivka**

```
obrazek.moveTo(100, 100);  
obrazek.curveTo(175, 125, 200, 200);
```



# Synchronizovaná multimédia – Flash

## ActionScript

- Grafika
- 4) kružnice

```
var size:uint = 100;  
// red circular shape  
obrazek.beginFill(0xFF0000);  
obrazek.moveTo(size / 2, 0);  
obrazek.curveTo(size, 0, size, size / 2);  
obrazek.curveTo(size, size, size / 2, size);  
obrazek.curveTo(0, size, 0, size / 2);  
obrazek.curveTo(0, 0, size / 2, 0);
```

více -> kreslicí rozhraní Flash API

[http://help.adobe.com/cs\\_CZ/ActionScript/3.0\\_ProgrammingAS3/](http://help.adobe.com/cs_CZ/ActionScript/3.0_ProgrammingAS3/)

# Synchronizovaná multimédia – Flash

## ActionScript

### - Multimédia – zvuk

```
import flash.events.Event;  
import flash.media.Sound;  
import flash.net.URLRequest;
```

```
var s:Sound = new Sound();  
s.addEventListener(Event.COMPLETE, onSoundLoaded);  
var req:URLRequest = new URLRequest("bigSound.mp3");  
s.load(req);
```

```
function onSoundLoaded(event:Event):void  
{  
    var localSound:Sound = event.target as Sound;  
    localSound.play();  
}
```

# Synchronizovaná multimédia – Flash

## ActionScript

### - Multimédia – video

```
var nc:NetConnection = new NetConnection();
nc.connect(null);
var ns:NetStream = new NetStream(nc);
ns.addEventListener(AsyncErrorEvent.ASYNC_ERROR, asyncErrorHandler);
ns.play("video.flv");
function asyncErrorHandler(event:AsyncErrorEvent):void
{
    // ignore error
}

var vid:Video = new Video();
vid.attachNetStream(ns);
addChild(vid);
pauseBtn.addEventListener(MouseEvent.CLICK, pauseHandler);
playBtn.addEventListener(MouseEvent.CLICK, playHandler);
stopBtn.addEventListener(MouseEvent.CLICK, stopHandler);
togglePauseBtn.addEventListener(MouseEvent.CLICK, togglePauseHandler);
function pauseHandler(event:MouseEvent):void
{
    ns.pause();
}
function playHandler(event:MouseEvent):void
{
    ns.resume();
}
```

# PCA – Principal Component Analysis

- **PCA** – **analýza hlavních** komponent; Karhunen-Loèveho (Hotellingova) transformace; Pearson (1901); Hotelling (1933)
- Lineární transformace původních dat na nové (nekorelované) - hlavní komponenty (rozptyl dat).
- Dekorelace dat; snížení dimenze vstupních signálů (vektorů čísel) při malé ztrátě informace
- Využití >>> komprese dat, redukce příznakového prostoru, rozpoznávání signálů, obrazů...
- Komprese RGB (multispektrálního) obrazu:
  - I) převod do stupně šedi  $I = 0.3R + 0.59G + 0.11B$  (komprese 33%)
  - II) využití PCA – vysoká korelace kanálů R, G, B použití 1. hlavní komponenty; nejvyšší zastoupení informace – obrázek s nejvyšším kontrastem

# PCA – Principal Component Analysis



$R$



$G$



$B$





# PCA – Principal Component Analysis

- 6) Výpočet vlastních čísel  $\lambda_i$  a vlastních vektorů  $e_i$

$$C e_i = \lambda_i e_i, \quad i = 1, \dots, M$$

- 7) Matice  $E_p$  ( $M \times M$ ) vytvořená z vlastních vektorů seříděných dle vlastních čísel:

$$E_p = [e_1 \dots e_M] \quad \begin{matrix} 3 \\ \square \\ 3 \end{matrix}$$

- 8) Vlastní prostor  $E$  je počítán z matic  $E_p$  and  $W$ :

$$E = W E_p$$

$$20\,000 \begin{matrix} 3 \\ \square \\ 3 \end{matrix} = \begin{matrix} 3 \\ \square \\ 3 \end{matrix} 20\,000$$

# PCA – Principal Component Analysis

- 9) komponenty:  $k_i = e_i + \bar{x}$  kde vlastní prostor  $E = [e_1 \dots e_M]$



$k_1$



$k_2$



$k_3$





# PCA – Principal Component Analysis



$k_1$



*gray*

# PCA – Principal Component Analysis

- **PCA** – analýza hlavních komponent, využití pro rozpoznávání
- $x_i$  jsou známé vektory příznaků pořízených ze signálu, obrazu...,  $s_i$  jsou jejich projekce do vlastního prostoru

$$s_i = E^T w_i$$

- $x_r$  neznámý vektor
- Postup rozpoznávání:
- 1) odečtení středního vektoru:

$$w_r = x_r - \bar{x}$$

- 2) projekce neznámého vektoru do vlastního prostoru

$$s_r = E^T w_r$$

- 3) rozpoznávání >>> porovnání neznámého vektoru  $s_r$  a známých vektorů  $s_i$ , např. na základě nejmenší vzdálenosti

# PCA – Principal Component Analysis

- **PCA** – analýza hlavních komponent, využití pro rozpoznávání - metriky
- Vzdálenost v městských blocích

$$d_B(s_r, s_i) = \sum_{m=1}^M |s_r(m) - s_i(m)|$$

- Euklidova vzdálenost

$$d_E(s_r, s_i) = \sqrt{\sum_{m=1}^M (s_r(m) - s_i(m))^2}$$

- Kosinová vzdálenost

$$d_C(s_r, s_i) = \frac{\sum_{m=1}^M s_r(m) s_i(m)}{\sqrt{\sum_{m=1}^M s_r(m)^2 \sum_{m=1}^M s_i(m)^2}}$$

- Mahalanobisova vzdálenost

$$d_M(s_r, s_i) = -\sum_{m=1}^M \frac{1}{\sqrt{\lambda_i}} s_r(m) s_i(m)$$

# PCA – Principal Component Analysis

- **PCA** – analýza hlavních komponent, využití pro identifikaci osob

