Low Bit-rate Video by Creation of Avatars for a Certain Class of Video Images – a VRML/Java Approach

Kunio Takaya
Department of Electrical Engineering,
University of Saskatchewan
57 Campus Drive, Saskatoon, Sask. S7N 5A9,
Canada
e-mail: takaya@engr.usask.ca

Naofumi Aoki
Research Institute for Electronic Science,
Hokkaido University
N-12 W-6 Sapporo, 060-0812, Japan
e-mail: naofumi.aoki@ma9.seikyou.ne.jp

ABSTRACT

The new capabilities added to MPEG-4 are the features of face animation and body animation. Low bit-rate video limited to such scenes that contain a face(s) is discussed in this paper in the light of those new features of MPEG-4. To aid in detection, analysis and tracking of a face, which are crutially imortant for such low bit-rate video coding, the Scott and Longuet-Higgins algorithm was studied to find the correspondence of feature points of a face.

I. INTRODUCTION

The speed of digital communication channels has been dramatically increased from the order of kbits/s for telephone to Giga bits in the internet backbones. It is not nowadays impossible to transmit digital video of 10 Mbits/s by using MPEG-2 via the satellite or optical fiber cables. The MPEG-2 is a coding scheme to compress video information down to 1/100 - 1/1000 of the original source volume, by means of the DCT (Discrete Cosine Transform) and motion vector search among the frames, (I-, P-, and B-frame) in the stream of video frames to eliminate intra- and inter-frame redundancies. The ordinary video in NTSC format requires at least 5 Mbits/s in digital video even with this high level of compression. The low bit-rate ersion of MPEG, called H-263, can compress a smaller CIF size image, and match the output bit stream to the speed of 64 Kbits/s achievable by digital telephone lines. In order to go beyond the limitation of such source coding methods, it will inevitably become necessary to extract only the useful and meaningful information contained in a video stream and transmit only the condensed abstracted information.

An avatar, synonym of a virtual clone, can be sent on behalf of a talking person to a receiving site, then the avatar can mimic the way the person talks for the remote audience, if the information such as head motion and facial expressions is provided in a descriptive text. Avatars have been created mainly for the virtual reality cyberspace as an agent of whoever wishes to explore the virtual 3-D world, such as a virtual town. A certain type of video scenes that contains a face, for instance, a news caster in the TV program, can utilize the concept of the avatar to compress the volume of video further beyond the capability of the traditional source coding. Creation of an avatar from a real person then to keep track of his motion and facial expressions are the challenge for the encoder.

Quoted from the article [4], MPEG-4 foresees that talking heads will serve an important role in future customer service applications. For example, a customized agent model can be defined for games or web-based customer service applications. To this effect, MPEG-4 enables integration of face animation with multimedia communications and presentations and allows face animation over low bit rate communication channels, for point to point as well as multi-point connections with low-delay. With AT& T's implementation of an MPEG-4 face animation system, we can animate a face models with a data rate of 300 - 2000bits/s. In many applications like Electronic Commerce, the integration of face animation and text to speech synthesizer is of special interest. MPEG-4 defines an application program interface for TTS synthesizer. Using this interface, the synthesizer can be used to provide phonemes and related timing information to the face model. The phonemes are converted into corresponding mouth shapes enabling simple talking head applications. Adding facial expressions to the talking head is achieved using bookmarks in the text. This integration allows for animated talking heads driven just by one text stream at a data rate of less than 200 bits/s. Subjective tests reported show that an Electronic Commerce web site with talking faces gets higher ratings than the same web site without talking faces. In an amendment to the standard foreseen in 2000, MPEG-4 will add body animation to its tool set thus allowing the standardized animation of complete human bodies

The same article also mentions that it is important to note that MPEG-4 only specifies the decoding of compliant bit streams in an MPEG-4 terminal. The encoders do enjoy a large degree of freedom in how to generate MPEG-4 compliant bit streams. In this paper, issues pertaining "low bit rate video by creation of avatars" are discussed from the viewpoint of translating a facial video to generate a face model, head motions, and associated facial expressions. As long as the bit stream produced by the encoder is compliant with the MPEG-4 (proprietary) face model,

This work was partly supported by a grant from Telecommunications Advancement Organization of Japan (TAO).

which specifies 84 feature points and 68 FAP (Facial Animation Parameters) categorized into 10 groups related to parts of the face, facial animation can be reconstructed and rendered at the receiving end. Encoding requires more work, as it requires detail feature analysis of the face.

II. 3D FACE MODEL

MPEG-4 handles multi-media objects such as still images, video objects, audio objects representing both natural and synthtic content types. Face animation and 2D mesh animation are included in MPEG-4, version 1 whereas body animation and 3D mesh animation are supported in version 2. Another functionality of importance is BISF (Binary Format for Scenes) which has made streaming multimedia contents, compression, and user interactivity possible. Thus, there are two ways to animate a face. One is to use the face model predefined in the face animation, then animate by means of FAP. The other is to use the BISF for model construction and manupilation. In either case, a face is modelled as a 3D wire frame mesh which is described by an indexed face set, *IndexedFaceSet* in VRML syntax. In the face animation, only those feature points defined by FDP (Face Definition Parameters) such as *bottom of the chin*, *left cheek bone* and *center of the pupil of left eye* must be supplied from the video source. As for motion of the face, general expressions such as *joy*, *sadness* or local movements such as *yaw_r_eyeball*, *dilate_l_pupil* and *raise_r_i_eyebrow* are among many other FAP's which manupilate facial expressions with respect to FDP. The feature points defined in FDP are shown in Fig. 1.

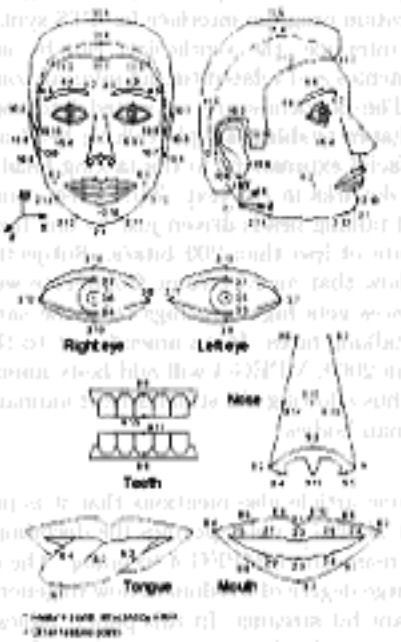

Fig. 1 Feature points in MPEG-4 face model

In order to take advantage of the face animation, it is extremely important to find FDP as well as FAP from a series of video frames to make an avatar look like a person in the video scene and act like that person. However, it is not so easy to automatically judge if there is a face or are more than one face in the scene, locate facial parts such as eyes, nose and mouth then keep track of their motions. In the next section, an approach originally proposed by Scott and Longuet-Higgins is applied to this problem. This method recognizes major features involved within a scene then tie successive images by one-to-one correspondence with respect to the detected feature points. Facial feature points that co-exist along with other features involved in the scene can be identified by the help of a knowledge base that defines the face. Once this step is accomplished, at least 2D FDP's that can be determined from a natural face looking straight ahead, are obtained. Closer local examinations are necessary with respect to eyes or mouth in order to determine specific values of FAP's.

III. FEATURE EXTRACTION AND MATCHING

Images in a video clip are highly correlated within a scene until panning the scene to a new scene occurs. Assume that at least one object is common to all the frames, for instance, a face. The ultimate objective here is to create an avatar of a human face contained in a series of video images as a 3D model. First problem is how to identify objects in a single frame. Second problem is how to correspond two objects of the same kind in between two frames. Dealing with human faces, it is apparent that motion of the head, opening and closing of mouth and eyes and the expression of emotion, all depend on successful feature identification and frame-to-frame matching.

When a video image is CIF size of 288×352 as shown in Fig.2, the image has a total of 101376 features, equivalent to the total number of pixels. This is, however, far too numerous compared with the number of FDP's. The shapes of the face and its components – eyes, nose, mouth, eye brows, ears are more important than texture, the outlines of the objects in a scene carry such information that object identification requires, that is, line drawing as opposed to painting contains more characteristic information as to find what objects are in the scene. Thus, for the purpose of feature extraction, each video image is first converted to an image of contour lines. Among many different ways to obtain contour lines of an image such as the gradient method, Laplacian filter, and multi-resolution sub-band filters to mention a few, the gradient method was used as the output represents gray scale variations for edges instead of the binary values.

Considering that the stronger is the gradient at edges, the better represents a feature, we can choose those points that exceed a certain threshold value as significant to represent the contour lines. This process of applying the gradient edge filter and thresholding reduces the number of features from the total number of pixels down to a small fraction of the image size. Although thresholding on the gradient filtered image detects important facial features,

it often fails to retain important but delicate minute details. This is contrary to the contour lines detected by the zero-crossing after the application of the Laplacian filter.

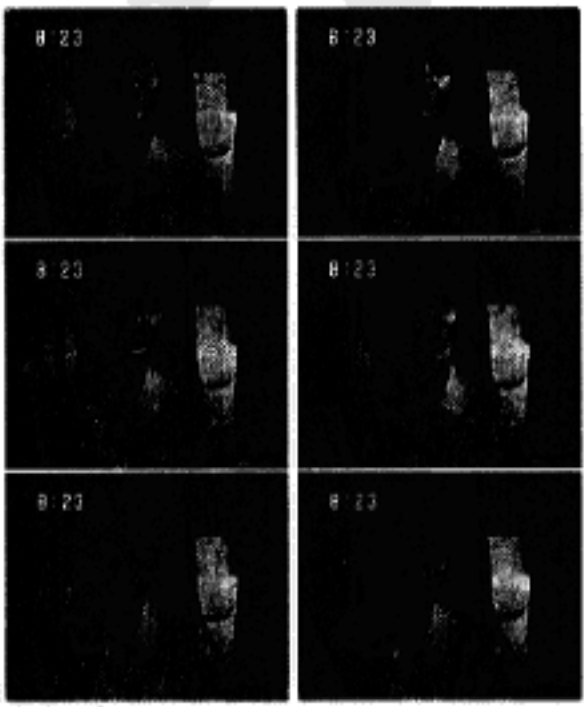

Fig. 2 Successive video frames at 3 fps, (moves from left to right then to the next row)

Thus, we obtain a sparse matrix representing the edges $E(m,n)$ having the gradient values only at those (m,n) where the gradient value exceeds the threshold. We further reduce the number of the element of $E(m,n)$ by selecting the largest N elements of $E(m,n)$. At the same time, we assign the value of 1 to non-zero points to produce a new sparse matrix of the exactly N features, namely $F(m,n) = 1mboxor0$. Following this process, the feature sparse matrices can be generated for successive video frames. Let $F_i(m,n)$ and $F_j(m,n)$ be the feature matrices of the i-th and j-th frame. Since both $F_i(m,n)$ and $F_j(m,n)$ take either 1 or 0, convert these to feature vectors $I(k)$, $k = 1,2,\cdots,N$ and $J(\ell)$, $\ell = 1,2,\cdots,M$. The next step is to associate feature points (or features in short) in the two vectors I and J which represent shapes or patterns in the images i and j. The feature points in both I and J represent similar patterns but not arranged in the same order. The method proposed by Scott and Longuet-Higgins [3] offers an excellent algorithm in associating the features of I with J.

The Scott and Longuet-Higgins algorithm is a straightforward application of the singular value decomposition (SVD) to a proximity matrix built from the feature vectors I and J. The proximity matrix $\mathbf{G}$ is a $M \times N$ matrix that defines Gaussian weighted distance between the feature vectors I and J, for which each element is given by

$$G_{k,\ell} = e^{-\frac{r_{k,\ell}^2}{2\sigma^2}}$$

$$r_{k,\ell} = ||I(k) - J(\ell)||.$$

Where, $k = 1,2,\cdots,M$, $\ell = 1,2,\cdots,N$ and $r_{k,\ell}$ is the Euclidian distance between the feature k of I and the feature ℓ of J. $G_{k,\ell}$ is a positive definite and monotonically decreasing with the distance $r_{k,\ell}$. σ controls the tightness of association. For a small σ, the features k and ℓ are regarded as associated when the distance between them is relatively small. Two points of a further distance are regarded as associated when a larger σ is chosen.

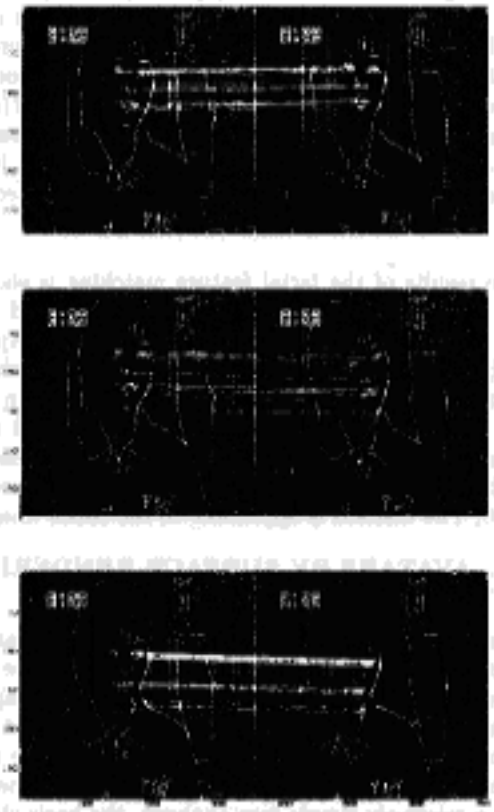

Fig. 3 Matching features between the two picures arranged side-by-side in Fig.2,

We now perform the singular value decomposition (SVD) on $\mathbf{G}$, that results in

$$\mathbf{G} = \mathbf{UDV}^T.$$

Where, $\mathbf{U}$ is a $M \times N$ matrix that consists of an orthogonal set of vectors which span the column space of the matrix $\mathbf{G}$. $\mathbf{V}$ is a $N \times N$ orthogonal matrix that spans the row space of $\mathbf{G}$. $\mathbf{U}$ and $\mathbf{V}$ are unitary and satisfy $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ and $\mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{I}$. When $M < N$, only the sub-matrix $M \times M$ of $\mathbf{V}$ is significant. $\mathbf{D}$ is a $N \times N$ diagonal matrix where a value of the diagonal corresponds to the magnitude of the dot product of the corresponding columns of the $\mathbf{U}$ and $\mathbf{V}$, i.e. singular value. The magnitude of the diagonal elements of $\mathbf{D}$ decreases as the column number increases. This means that the first columns in the $\mathbf{U}$ and $\mathbf{V}$ are the major contributor of the matrix $\mathbf{D}$

(principal orthogonal vectors).

The Scott and Longuet-Higgins principle asserts that

$$\mathbf{P} = \mathbf{UEV}^T$$

obtained by replacing **D** by the identity matrix **E** yields the association between the features I and J. Instead of letting a few principal vectors in the **U** and **V**, which correspond to the major singular values $D_{k,k}$, represent the proximity matrix **G**, the identity matrix **E** rather forces all the orthogonal vectors to evenly participate to produce **P**. If the element $P_{k,\ell}$ of **P** is the greatest element in its row and the greatest element in its column, the feature $I(k)$ and the feature $J(\ell)$ are regarded as 1:1 correspondence each other. If this is not the case, the feature $I(k)$ unsuccessfully competes with other features for partnership with $J(\ell)$. Also, the property that $\sum_\ell P_{k,\ell}^2 = 1$ implies that a feature $I(k)$ cannot be strongly associated with more than one feature $J(\ell)$.

The results of the facial feature matching is shown in Fig. 3. The number of feature points presented to the Scott and Longuet-Higgins algorithm is 100. Approximately 65% of the 100 feature points were matched for this particular set of video images. As shown in Fig.3, good matching are found for the eyes, nose and mouth between the two images. Though this experiment used only the face part to keep the number of feature points to be 100, The method is applicable to the entire images.

IV. AVATARS BY SURFACE RENDERING WITH TEXTURE

The face animation of MPEG-4 is intended to animate a synthetic face. Although the face model can be modified to resmble the orignal face in terms of its shape and the texture (2D image), the created avatar is not a real replica of the person. An alternative to the use of this face model is to use BIFS (Binary Format for Scenes). By using a relatively simple face analyzer, the basic shape of his/her head can be determined from a few feature parameters. By means of a face analyzer program that measures horizontal-vertical ratio, deviation factor from an ellipse, chin ratio, nose position/ratio, eye position/ratio, a wireframe model of the face can be constructed automatically. Then, the texture of the 2D source image is superimposed on the wireframe. The temporal changes of facial expressions are supplied to the avatar as a sub-picture of the face alone. The size of the sub-picture must be equal to the mesh size covering the face area of the avatar. The encoder simply keeps track of head motion and disregards the face, in other words, the movements of facial components altogether. BIFS can create avatar's wireframe from the basic face parameters. There are mechanisms in BIFS to send sub-pictures as texture, then to move the avatar's head. Figure 4 shows an avatar constructed this way. To make the images a little more realistic, the wireframe mesh around the nose was deformed from the egg shaped.

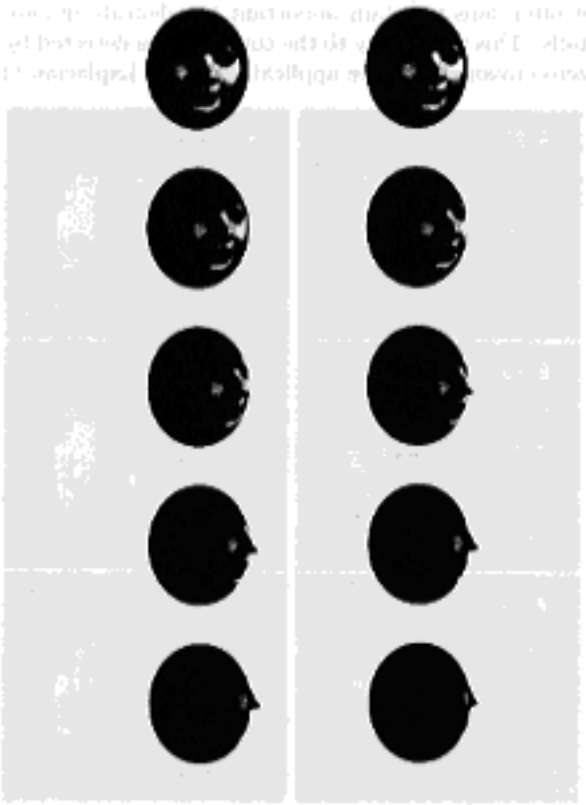

Fig. 1 Head rotation of the texture based avatar, (moves from left to right then to the next row at 3 fps)

V. VRML AND JAVA SCRIPT

As mentioned in the article [5], while an MPEG-4 BIFS scene has, for the most part, a structure inherited from the Virtual Reality Modeling Language (VRML 2.0), its explicit bit stream representation is completely different. Moreover, MPEG-4 adds several distinguishing mechanisms to VRML: data streaming, scene updates and compression. In MPEG-4 using a client-server model, An MPEG-4 client (or browser) contacts an MPEG-4 server, asks for content, receives the content, and renders the content. This 'content' can consist of video data, audio data, still images, synthetic 2D or 3D data, or all of the above.

As far as the interactive data streaming is concerned, VRML and Java script can jointly perform the same functionality. Since both the face model in the face animation and the wireframe mesh in BIFS define a 3D object by *IndexFaceSet* of VRML, construction of a very simplified avatar and motion control are demonstrated by using VRML instead of MPEG-4. The following code constructs and renders the facial components, eyes, nose and mouth, then moves the eyes and the lips synchronized with timer, smoothly by interpolation applied to the indexed face set.

```
Transform {
  translation 1.5 0 0
  children [
    DEF head Shape{
      appearance Appearance {
    material Material {
      diffuseColor 1 0.7 0.6
    }
      }
      geometry IndexedFaceSet {
        coord DEF f Coordinate {
      point [ -1 0 0,   1 0 0,  0 0.5 0 ,  0 1 0,  0 -0.6 1,  0 -1.2 0 ]
    }
        coordIndex [0 1 2 -1   0 2 3 -1   1 2 3 -1
                    0 1 4 -1   0 4 5 -1   1 4 5 -1 ]
    solid FALSE
      }
    }
  ]
}
Transform {
  translation -1.5 0 0
  children [
    USE head
  ]
}
Transform { translation 1.5 0 0
  children [
    DEF eyeball1 Shape {
      appearance Appearance {
    material Material {
      diffuseColor 1 1 1
      transparency 0.8 }  }
      geometry Sphere { radius 0.5 } }
  ] }
Transform { translation 1.5 0 0
  children [
    DEF iris Shape {
      appearance Appearance {
    material Material {
      diffuseColor 0.1 0.9 0.9 }}
      geometry Sphere { radius 0.2 }}
    ] }
Transform {
  translation -1.5 0 0
  children [
    USE eyeball1 ]
}
Transform {
  translation -1.5 0 0
  children [
    USE iris ]
}
Transform {
  translation 0 -3 0
  scale 1.0 0.8 0.7
  children [
    DEF beak Shape{
      appearance Appearance {
    material Material {
      diffuseColor 1 0.3 0.3
    }
      }
      geometry IndexedFaceSet {
        coord DEF f2 Coordinate {
      point [ -1 0 0,   1 0 0,  0 0.5 1 ,  0 1 0,  0 -0.6 1,  0 -1.2 0 ]
    }
        coordIndex [0 1 2 -1   0 2 3 -1   1 2 3 -1
                    0 1 4 -1   0 4 5 -1   1 4 5 -1 ]
    solid FALSE
      }
    }
  ]
}
Transform {
  translation 0 -1 0
  children [
    DEF nose Shape{
      appearance Appearance {
    material Material {
      diffuseColor 0.8 0.4 0.4
    }
      }
      geometry IndexedFaceSet {
        coord DEF f3 Coordinate {
      point [ -0.5 0 0,   0.5 0 0,  0 0 1 ,  0 1 0 ]
    }
        coordIndex [0 1 2 -1   0 2 3 -1   1 2 3 -1
    solid FALSE
      }
    }
  ]
}
DEF TIMER TimeSensor {
    cycleInterval 1.0
    loop TRUE
}
DEF C12 CoordinateInterpolator {
    key [0.0 0.4]
    keyValue [ -1 0 0,   1 0 0,  0 0.5 1 ,  0 1 0,  0 -0.6 1,  0 -1.2 0
               -1 0 0,   1 0 0,  0 0 1 ,  0 0.6 0,  0 -0.2 1,  0 -1 0 ]
}
ROUTE TIMER.fraction_changed TO C12.set_fraction
ROUTE C12.value_changed TO f2.set_point
DEF C1 CoordinateInterpolator {
    key [0.0 1.0]
    keyValue [   -1 0 0,   1 0 0,  0 0.5 1 ,  0 1 0,  0 -0.6 1,  0 -1.2 0
                 -1 0 0,   1 0 0,  0 0 1 ,  0 0.6 0,  0 -0.2 1,  0 -1 0 ]
}
ROUTE TIMER.fraction_changed TO C1.set_fraction
ROUTE C1.value_changed TO f.set_point
```

Fig. 6 3D Avatar created by the VRML script shown

VI. CONCLUSION

The fesibilty of the concept to have the avatar represent a person in video scenes is studied in this paper. The issues related to identifying facial feature points and matching the same features are discussed in depth. The Scott and Longuet-Higgins algorithm is found to be promising for automatically locating those features and also for the purpose of motion tracking. MPEG-4 has all tools needed by the low bit rate video utilizing the avatar (agent) resembling a person in video scenes. However,building an intelligent encoder requires more sophisticated knowledge base even with the Scott and Longuet-Higgins algorithm for video image analysis.

REFERENCES

[1] Kunio Takaya, Li Ding and Naofumi Aoki, "H-263 Facial Video Image Coding with the MELP Voice Codec", Proceedings of IWSCI'99 IEEE International Symposium on Soft Computing for Industries 1999, Muroran, Japan, June 1999.

[2] M. Pilu "A direct method for Stereo Correspondence based on Singular Value Decomposition", IEEE International Conference of Computer Vision and Pattern Recognition, Puerto Rico, June 1997.

[3] G. Scott and H. Longuet-Higgins, "An Algorithm for associating the features of two patterns", In Proc. Royal Society London, volume B244, pp.21-26, 1991.

[4] Murat Tekalp and Jorn Ostermann, "Face and 2-D Mesh Animation in MPEG-4", Tutorial Issue on The MPEG-4 Standard, Image Communication Journal available at MPEG-4 Industry Forum, http://www.m4if.org/resources.html

[5] Julien Signes, Yuval Fisher and Alexandros Eleftheriadis, "MPEG-4's Binary Format for Scene Description", Tutorial Issue on The MPEG-4 Standard, Image Communication Journal available at MPEG-4 Industry Forum, http://www.m4if.org/resources.html