

## INTERNET ENABLED DIGITAL PHOTOGRAPHY

Peter M. Corcoran<sup>1</sup>, Petronel Bigioi<sup>1</sup>, Eran Steinberg<sup>2</sup>, and Yury Prilutsky<sup>2</sup>

<sup>1</sup>Dept. of Electronic Engineering, University College, Galway

<sup>2</sup>Fotonation Inc., San Mateo, CA

### Abstract

The principles and concepts of Internet Enabled Digital Photography are outlined. The design and implementation of an interface appliance to connect digital cameras to the Internet is described. In addition details are given of the server-side infrastructure required to support a range of network-oriented services.

### 1. Introduction

The digital camera is one of the most successful new consumer electronic products of recent years. Indeed there has been such explosive growth in today's marketplace that there are now more than 30 manufacturers of competing digital still camera (DSC) products. However, although many digital cameras are functionally similar each camera is accessed via its own proprietary software and protocols on the user's personal computer.

As we approach the next millennium it is clear that a new generation of digital cameras is emerging. These new products are increasingly sophisticated and are rapidly superceding conventional photography. However a digital camera is still tied to a home or office computer, and indeed to a proprietary software package on that computer. This limits both the utility and the target user market of the digital camera. If digital cameras are to reach their full potential as consumer devices then they must be freed from this dependence on a personal computer.

In an earlier paper [1] we consider some of the issues involved in the design of a general Java API for digital cameras. We now consider how a new generation of intelligent home appliances would be capable of providing access to and functional integration with the Internet using just such a generic camera API.

In this paper we describe how today's digital cameras can be integrated with the Internet by the addition of an external embedded interface appliance. A prototype implementation of such an appliance is described. In many respects this is a simplified form of the standard desktop PC. By using an embedded CPU which is compatible with many desktop products it is possible to leverage existing interface software developed for digital

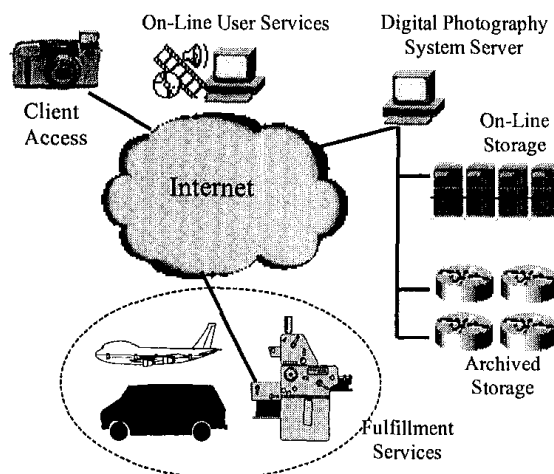
camera applications and to reduce the unit cost to consumer levels.

### 2. Elements of Internet Photography

In this section we give an overview of the key infrastructural elements of an Internet Photography (IPh) system. These are summarized in *fig 1* and comprise (i) a client-access mechanism; (ii) a server-infrastructure and (iii) fulfillment services.

The client-access mechanism provides a means to reliably download pictures to an on-line server which provides storage, access and management services for digital pictures. The pictures are maintained on-line for a certain period and then migrated to archival storage for long-term storage. Users can access pictures stored on-line and a range of services are available to manage and post-process images. Finally, volume fulfillment services bring the cost of hard-copy pictures into the same cost bracket as conventional photography.

Each of these IPh system elements is now discussed in more detail.



*Fig 1: The Key Elements of Internet Photography.*

#### 2.1 Client Access

The Internet Photography (IPh) client differs from most other typical Internet clients. Normally the key function of a client is to access data on the Internet. Thus the

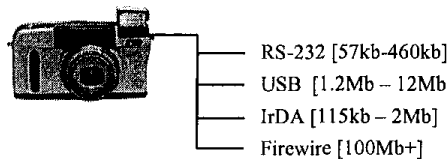
client draws information and data from a remote server – typically a Web or HTTP server. In the case of an IPh client the key function is to load data from the client to the server thus reversing the conventional sense of bulk data flow in a client-server system.

This reversal of the sense of bulk data flow introduces some interesting perturbations of the normal client-server data model. These will be discussed in due course. For now we will focus on the principle functional requirements of the client-access element of our IPh system.

### 2.1.1 Interfaces to Digital Cameras

In this section we cover the physical mechanisms of accessing digital picture data stored on a typical digital still camera (DSC). Most of today's cameras use an RS-232 compatible physical layer. However the newer cameras are adopting faster physical interfaces such as IrDA and USB. Ultimately, as the size of CCD arrays and improvements in onboard image post-processing increases the size of digital pictures beyond one Megabyte, most cameras will adopt IEEE 1394, or Firewire as their principle physical interface.

The main physical mechanisms of communication with a typical DSC and their representative rates for data transfer are shown in *fig 2*. Note the significant advantage that Firewire has in terms of data transfer rates.



*Fig 2: Physical Interfaces to a Digital Still Camera.*

Each physical interface must support additional software protocol layers to allow interaction between the camera and its outside environment. Typically all digital cameras to date have required a PC in order to download, print or access the pictures. Recently a number of new camera appliances have appeared which allow pictures from a camera to be printed directly when combined with an appropriate printer.

### 2.1.2 Internet-Integration Appliance (IIA)

In this section we describe how today's digital cameras can be integrated with the Internet by the addition of a simple embedded interface appliance. Such an appliance can be very lightweight and portable - much more so than a conventional laptop, or even a handheld PC. More

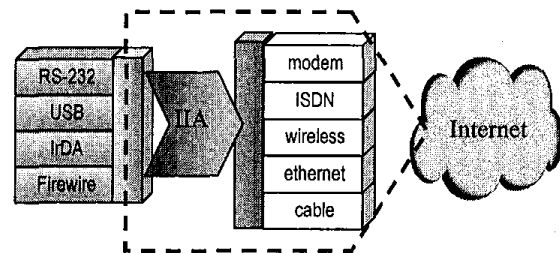
importantly such a dedicated appliance does not need a particular software package to interface with a particular camera, but can connect to any generic digital camera.

Such an Internet-Integration (IIA) appliance also opens the doors to a new generation of server side Internet services and applications based on digital photography.

A prototype implementation of such an appliance and the associated system software is described in sections 3.1 and 3.2 of this paper.

In many respects this appliance is a very simplified form of the standard desktop PC. By using an embedded CPU that is compatible with many desktop products it is possible to leverage existing software and protocol modules developed for digital camera applications [1] and thereby reduce the unit cost of the IIA to consumer levels.

An overview of the functional requirements for a generic IIA appliance is summarized in *fig 3*. For the prototype implementation described later in this paper we have confined ourselves to the commonest means of accessing pictures on a digital camera – via the RS-232 port – and the commonest means of establishing an Internet connection – via the telephone network using a conventional analog modem.



*Fig 3: Connectivity Requirements for a generic IIA.*

### 2.1.3 Client-Server Models for Internet Photography

Mention was already made of the difference between the IPh client-server model and more conventional client-server models. The IPh client is, in a sense, a micro-server and this introduces some complications that are unique to the IPh client-server model. Furthermore the modem version of the IPh client suffers from bandwidth limitations which limit the data throughput to the main system server.

To best understand these issues it is useful to study the protocol stack shown in *fig 4* below. The key point is that, for a modem-based version of the IPh client hardware there are PPP protocol layers in addition to the standard TCP/IP protocol layer. The PPP guarantees robust communications, but when large block of digital

data, such as digital pictures are transmitted through such a protocol stack there is a significant overhead. In our best-case studies between 30% and 40% of the transmission bandwidth is lost to such protocol overheads. Note that there is less overhead in an Ethernet based stack.

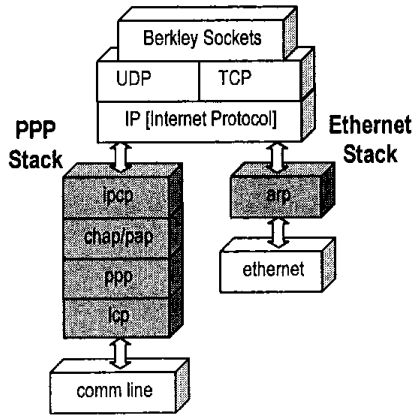


Fig 4: Protocol Stacks for IPh Client.

This form of IPh client has a complete TCP/IP solution integrated into the basic IPh hardware. Thus the IPh client connects a digital camera directly to the Internet over the modem/PPP link and digital pictures are sent directly to the main system server over this TCP/IP link. We call this a two-tier client server solution and the main system infrastructure is illustrated in fig 5 below.

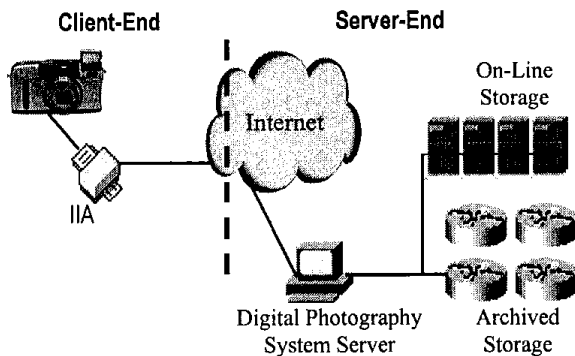


Fig 5: Two-Tier Internet Photography Infrastructure.

There are a number of key drawbacks to this two-tier architecture. Apart from the protocol overhead we must also consider that if the IPh client does not succeed in making a direct, fast connection to the remote system server then there will be additional delays over the Internet connection. This will increase the online time required by the client and if a link proves unreliable then

pictures can easily be lost in transit from IPh client to the main server.

A more complex three-tier system architecture overcomes these difficulties and provides a more reliable infrastructure. It also requires a less complicated IPh client and allows faster data transfer from the IPh client to the Internet. This alternative systems architecture is illustrated in fig 6 below.

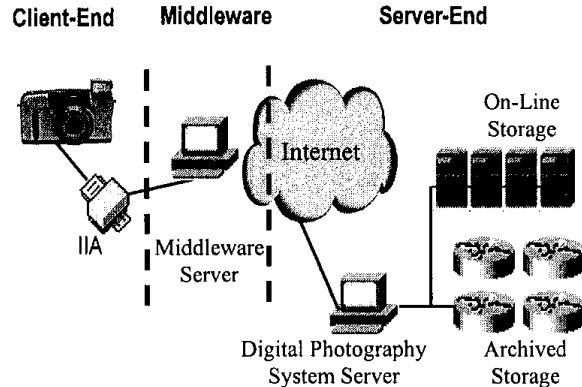


Fig 6: The Three-Tier Infrastructure.

### 2.2 Server-Side Infrastructure

To support the client appliance it is important that a flexible server-side infrastructure exists to provide the end user with a range of exciting and interesting new services, focussed around the network-enabled digital camera.

The key services that will initially be required in order to promote the success of digital photography on a mass-market scale are:

- (i) picture fulfillment services
- (ii) on-line picture storage
- (iii) archival picture storage
- (iv) added-value on-line services

In an experimental laboratory set up these services can be adequately provided by a conventional PC-based web server with a large hard disk to provide on-line storage and a tape drive to provide archival storage facilities. Added value services can be programmed using one of the high-level scripting languages that are popular for web development.

Note that a key aspect of this client-server architecture for digital photography is that practically all of the client configuration information can be maintained on the server-side. This, in turn, reduces maintenance issues and facilitates upgrading of services and the infrastructure as a whole.

### 2.3 Fulfillment Services

As the system described in this paper is an experimental set-up we do not offer a full commercial fulfillment service. However such a service would be based on the equivalent fulfillment service for conventional photography. Such a service allows the cost-per-print of a photograph to be reduced to the order of several cents – an important factor for the consumer who is likely to be discouraged from adopting digital photography by the cost of home-based color prints.

An important factor here is that as no developing or handling of negatives is necessary for digital photographs there will be significant simplifications in the procedures for processing digital photographs. This, in turn, is likely to drop unit costs per picture below those of conventional photographs once the market volume for digital photography reaches 5-10% of the conventional photography market.

### 2.4 On-Line User Services

One of the key advantages that digital photography has over conventional photography is the inherent flexibility of the digital medium. Thus it is a good bet that there will be a broad range of new added-value services developed to compliment the basic system functionality of picture upload, storage and fulfillment.

In this section we briefly describe some of the preliminary on-line services which we consider the most important. Functional versions of most of these services have also been prototyped in our experimental setup.

#### 2.4.1 Verification Services

This service is to allow the user to confirm that their pictures have been successfully uploaded to the main photography server. This may be achieved in several ways, one of the simplest being to allow the user to browse their photographs using a standard web browser. Each user has a private password to allow them access only to their personal pictures.

Note, however, that not all users will have a computer. (Recall that one of the goals of our project is to eliminate the need for a PC in order to use a digital camera). In such cases it would be possible to fax back or post a contact sheet with thumbnail images of the uploaded pictures. This confirms to the user that their pictures have been successfully “developed” and also allows them to choose which pictures they would like to order in hard-copy.

#### 2.4.2 Access and Management Services

Once a user has uploaded a “digital film” to the server they may wish to sort and re-order these “digital negatives”. This can be achieved either by running server-side scripts or using Java or Active-X modules to re-arrange and sort picture files as if they were on a local hard disk.

Note that one of the advantages of keeping pictures on a dedicated photography server is that the original “digital negatives” are preserved and thus some of the permanence of conventional film negatives is achieved. Original pictures are date and time-stamped in a database and any re-sorting of pictures in a user’s directory is achieved by cross-linking to the original image.

#### 2.4.3 Touch-Up Services

There are quite a few programs available which offer various forms of post-processing of digital images. Many of these offer capabilities to sharpen images that are poorly focussed, to remove artifacts such as red-eye and to optimize the color balance of images. The functionality of such services can be offered over a web interface allowing the user to adjust and modify images secure in the knowledge that the original “digital negative” will be preserved and can be restored if desired.

#### 2.4.4 Ordering Services for Hard-Copy Pictures

Of course the most important service for most users will be the ordering and fulfillment service. This should be available through both an on-line web interface, but also through a mail-order and/or fax ordering service for users who may not own their own PC.

Naturally users should be billed for usage of all of these services. Thus every time a user re-sorts their picture albums or modifies an image their account will be charged. Pictures would normally only stay on-line for a limited time period, after which they will be migrated to archival storage. However they may be restored at any time, or favourite photographs can be kept on-line in an active-album folder.

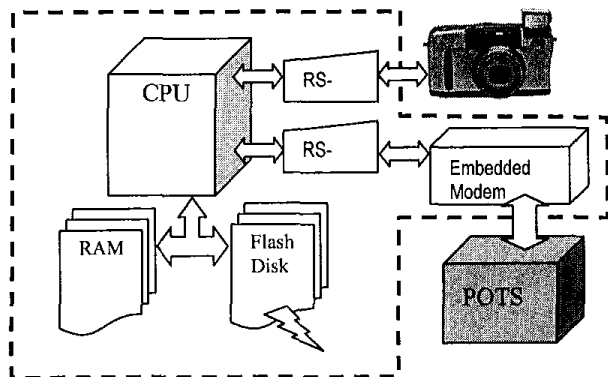
## 3. Prototype System Description

In this section we give some details of our experimental implementation of an internet-enabled digital photography system. Some descriptions of the system services and overall system infrastructure have already been given, so here we will focuss on practical implementation issues.

### 3.1 The IIA Hardware Implementation

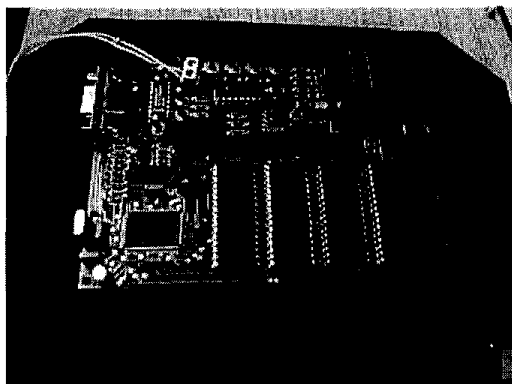
Here we give an overview description of our actual IIA client hardware. As was mentioned earlier, this hardware is limited to RS-232 camera connectivity and to analog-modem Internet connectivity. The Internet connectivity is achieved either over a PPP link or via a direct, non-IP link to a middleware server, which provides an internet relay service.

The layout of the hardware is illustrated in *fig 7* below. The system has both RAM and flash memory. The flash has a boot area, which can be dynamically reprogrammed, and a disk area, which appears like a DOS compatible disk to the operating system.



*Fig 7: Client Hardware is based around a x86 CPU*

In *fig 8* we show a photograph of the original IIA prototype. Note the embedded modem on the right-hand side of the PCB at the back. An idea of scale can be gotten from the RS-232 connector on the left.



*Fig 8: An initial prototype of the client IIA hardware.*

### 3.2 Overview of System Software

The main components of the system software are the embedded software running on the IIA client and the

internet-photography server software. These are described below.

#### 3.2.1 Client-End Software

The IIA client runs an embedded OS that is compatible with DOS. Depending on the client-server architecture the IIA client may run either a TCP/IP stack and PPP or it may run a more lightweight non-IP data transfer program.

From our experiences there is significant overhead in implementing a full TCP/IP stack on the client. Furthermore as TCP/IP does not, in its present form, guarantee any quality of service (QOS) it is conceivable that picture download over a slow link will be both time-consuming and costly for the consumer. If a lightweight non-IP transfer program is used then practically all the limited modem bandwidth can be devoted to transferring picture data. This offers a 30-40% performance improvement over the best case for TCP/IP.

The client also runs an embedded camera protocol module with establishes a connection with the camera and downloads the “digital film” from the camera. The IIA can be easily reprogrammed with a wide range of camera protocols, or these may be loaded dynamically from the internet-photography server.

#### 3.2.2 Server-End Software

The core server software will depend on the system architecture. If the client supports a full TCP/IP stack then pictures are basically sent from the client using the internet e-mail protocol, SMTP. The main server program listens at the appropriate SMTP port and when a connection with the correct format – including a valid client ID – is made then the attached pictures are stored in that users data space.

If the three-tier architecture is implemented then a middleware server should run a specialized non-IP daemon, which waits for a modem connection to be made. When such a dial-up connection is established the program accepts the upload of pictures with CRC error checking. These pictures are then queued, with the client ID, for delivery over SMTP to the main system server.

### 3.3 A Basic Picture-Album Application

In this section we describe the basic user interface to our internet-photography server. This can be seen in *fig 9* below. This initial user interface is quite basic but provides the end-user with most of the services we described earlier.

The right-hand frame displays thumbnails of each picture and if a thumbnail is clicked the full-size image appears in the main frame of the browser. Thus the user can browse and verify his pictures. This current interface does not support on-line management and re-sorting of photo albums, although we have created some Active-X controls, which allow this to be achieved on Windows-based systems.

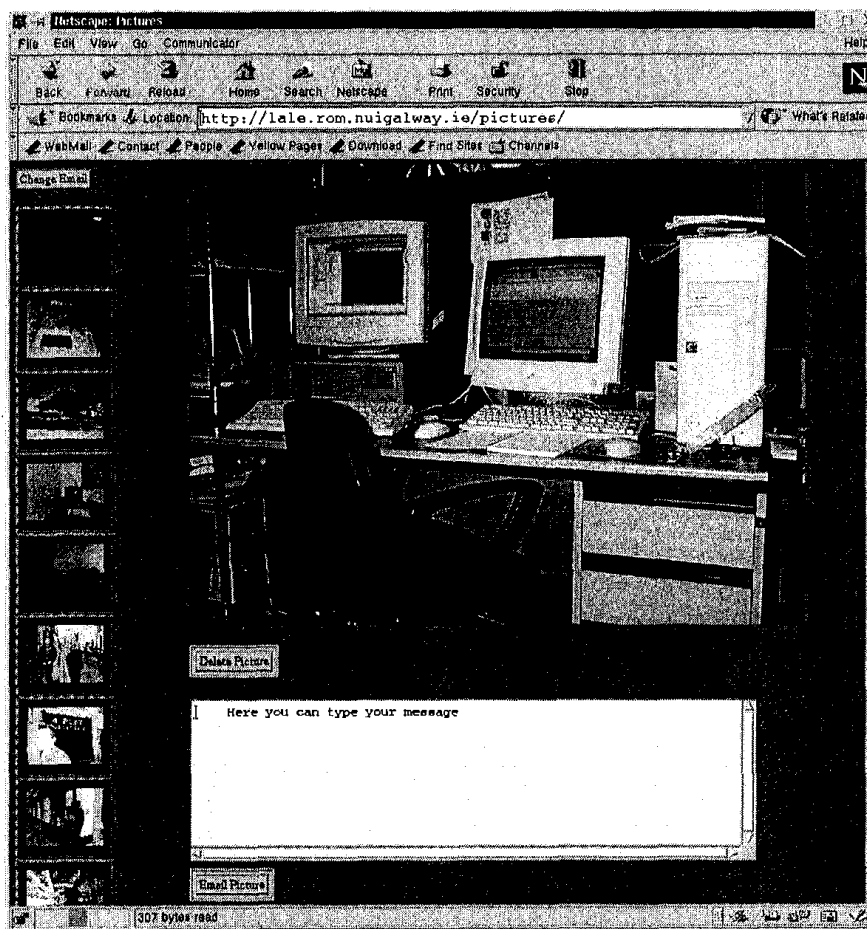
There is also an e-mail facility which allows the end-user to select a picture and to mail it to someone.

#### 4. Conclusions & Future Work

In this paper we have demonstrated that the infrastructure for internet-enabled digital photography is

available. Our main goal has been to simplify user access to digital cameras by automating the upload of pictures and providing users with internet-based tools and applications to access and manage their pictures.

Ultimately we predict that the functionality of the client-end appliance will be directly integrated into the digital camera itself. This will simplify the use of digital cameras from the consumer's perspective. We believe that the system described in this paper provides a practical bridge between the digital cameras available in today's market and future models which will, without doubt, incorporate embedded Internet access directly in the camera unit itself.

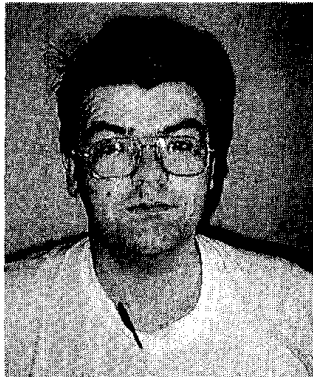


*Fig 9: On-line digital picture album.*

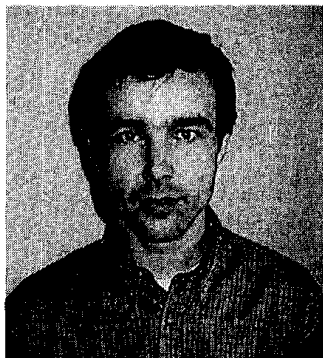
## REFERENCES

- [1] Corcoran P.M., Papai F., Zoldi A., and Desbonnet J., "A Portable Java API Interface to Simplify User Access to Digital Cameras", *IEEE Transactions on Consumer Electronics*, Vol. 44 No. 3 p686-691 Aug. 1998.
- [2] Corcoran P.M., "Mapping Home-Network Appliances to TCP/IP Sockets Using Three-Tiered Home Gateway Architecture", *IEEE Transactions on Consumer Electronics*, Vol. 44 No. 3 pp729-736 Aug. 1998.

## BIOGRAPHIES



Peter Corcoran received the BAI (Electronic Engineering) and BA (Maths) degrees from Trinity College Dublin in 1984. He continued his studies at TCD and was awarded a Ph.D. in Electronic Engineering for research work in the field of Dielectric Liquids. In 1986 he was appointed to a lectureship in Electronic Engineering at UCG. His research interests includes microprocessor applications, environmental monitoring technologies. He is a member of I.E.E.E.



Petronel Bigioi received his B.S. degree in Electronic Engineering from "Transilvania" University Brasov, Romania, in 1997. At the same university he received in 1998 M.S. degree in Electronic Design Automation. Currently he is completing his M.S. degree in Electronic Engineering at University Collage Galway, Ireland. His research interests include VLSI design, communication network protocols and embedded systems.