



Image Processing over IP networks (IPoIP™)

White Paper

By Zvika Ashani / CTO Agent Vi Inc.

Table of Contents

1	Overview	3
2	The Existing Methods of Remote Automated Image Processing	3
3	Requirements for a Viable Solution (Technically and from a Cost Perspective).....	5
4	IPoIP™ Architecture.....	6
5	Applications in the Physical Security Market	9

1 Overview

The growth of the Electronic Media, of Process Automation, and especially the outstanding growth of attention to National and Personal Security in the past few years have all contributed to the growing need of being able to automatically detect features and occurrences in pictures and video streams on a massive scale, without the need for human eye intervention and in real time. To date, all technologies available for such automated processing have come short of being able to supply a solution that is both technically viable and cost-effective.

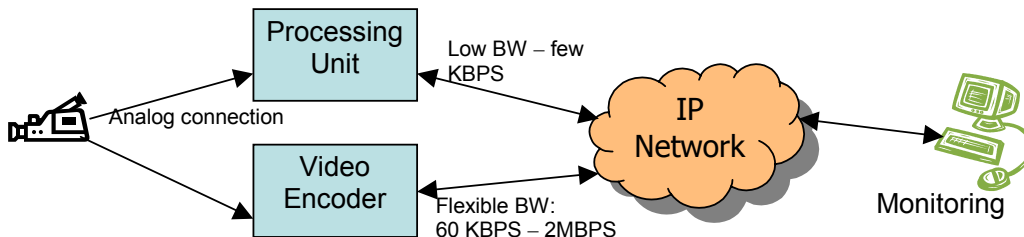
This white paper details the basic ideas behind a novel, patent-pending technology called Image Processing over IP networks (**IPoIP™**). As its name implies, **IPoIP™** provides a solution for automatically extracting useful data from a large number of simultaneous image (video or still) inputs connected to an IP network, but unlike other existing methods, does so at reduced costs without compromising reliability. The document will also outline the existing image-processing architectures and compare them to **IPoIP™**.

Ending this document will be a short chapter detailing several possible implementations of **IPoIP™** in existing applications.

2 The Existing Methods of Remote Automated Image Processing

A tremendous amount of research effort has been put into the ability to extract meaningful data out of captured images (both video and still) in the past years. As a result, a large number of proven algorithms exist both for real-time and offline applications, algorithms that are implemented on platforms ranging from pure software to pure hardware. These platforms, however, are generally designed to deal with a relatively small number of simultaneous image inputs (in most cases actually no more than one). They are designed in one of two main architectures.

2.1 “Local processing” architecture



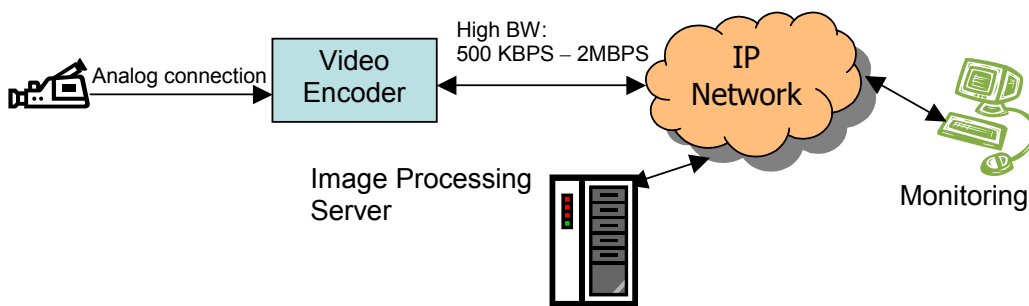
This is by far the most commonly available system architecture for image processing. The main idea behind it is that all the processing is done at the camera location by a processing unit, and the results are then transmitted through a network connection to the monitoring area. The processing unit is usually PC based for the more complex solutions but the recent growing trend is to move the

processing to standalone boxes based on a DSP or even an ASIC. It performs the entire image-processing task and outputs a suitable message to the network when an event is detected. Also residing at the location of the camera is a video encoder that is used for remotely viewing the video through the IP network. It can be configured to transmit the video at varying qualities depending on the available bandwidth. The video is transmitted using standard video compression techniques such as MJPEG, MPEG-4 and others.

When cost is less of an issue, this architecture provides an adequate solution for running a single type of algorithm per camera. However, when the number of cameras increases and a more robust solution is needed (which is in many times the case), this solution falls short due to the following reasons:

- Each camera requires its own dedicated processing resources, causing the system cost to scale linearly with the number of cameras needed. No cost reduction is possible when dealing with a large-scale system.
- Each additional type of algorithm requires additional processing resources and integration between various algorithms is costly.
- In case of cameras that are distributed outdoors, PC based products provide an inadequate solution due to space limitations and their inability to withstand harsh environmental conditions.
- DSP based solutions require a much higher development effort because of limited resources and inferior development tools.

2.2 “Server processing” architecture



The second type of system architecture (although far less common) is the “Server Processing” architecture. All of the image processing tasks are put on one single powerful server that serves many cameras. From a hardware point of view, this solution is more cost effective and is suitable for large-scale deployments. This architecture is made possible due to the fact that there are only a small percentage of “interesting” occurrences in each camera, requiring only a small amount of actual processing power and allowing for one server to deal with many cameras.

Where this architecture comes short is on the network side – it has extraordinary bandwidth requirements. Because all of the image-processing functions are performed at the server, it needs to receive very high quality images in order to

provide accurate results. This creates a need for significant network resources. When the application runs on a LAN with a relatively small number of cameras this may be possible, but for distributed applications with large numbers of cameras the solution becomes impractical because of the costly network infrastructure required. This also leads to the fact that this type of architecture is usually used in applications where the algorithm works on a single frame at a time and not on a full video stream.

3 Requirements for a Viable Solution (Technically and from a Cost Perspective)

Having understood the limitations of the existing image processing architectures, let us now look at the requirements for a cost-effective and technically viable solution.

Such a system must have the following characteristics:

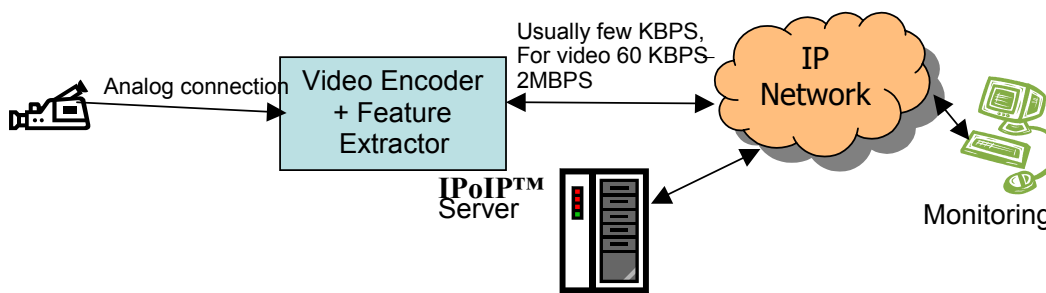
- Scalability and mass-scale abilities – the system must be able to handle deployments ranging from a few dozen cameras up to thousands of cameras simultaneously.
- Scalability from a cost perspective – no matter what the scale of the deployment is – the system has to provide a cost-effective solution.
- The cameras should be able to be installed in geographically remote locations (under the assumption that there is an IP network connection to these locations).
- It must be possible to view each camera remotely from a monitoring station connected to the network.
- One or more image processing algorithms needs to be applied to each camera at any given moment. The outputs of these algorithms need to be collected in a central database and should also be viewable on the monitoring station.
- It should be possible to easily add new algorithms or customize existing ones without requiring massive upgrades to the system.
- There's a need to detect both single-camera events and multi-camera events. Multi camera events fuse the information from several sensors to create a higher level event.
- In Rural areas (tracks, pipelines, borders) where there's no infrastructure, power requirements and bandwidth (especially if using wireless) are very important. For these types of installations where power consumption is critical, installing PC's is not an option.

4 IPoIP™ Architecture

The **IPoIP™** architecture was designed to answer the needs defined above with the following key goals in mind:

- Providing a cost effective solution for image processing applications over a large number of cameras without sacrificing detection probability or increasing false alarm rate (FAR).
- Enabling the application of any algorithm to any camera even if it is in a geographically remote location with limited supporting facilities.
- Providing the ability to apply a wide range of algorithms simultaneously to any camera without limiting the user to only a single application at a time.

The uniqueness of **IPoIP™** is a distributed image processing architecture. Instead of performing the image-processing task either at the camera or in the monitoring area using one of the two aforementioned architectures, the algorithms are performed in both locations. They are segmented into 2 parts and divided between the video encoder hardware and the central image processing server. In this way **IPoIP™** is able to retain the strengths of both the “Local” and “Server” architectures, while avoiding their limitations.



Note: the term *Video Encoder* refers to any device capable of compressing video and streaming it over an IP network. This can be a DVR a video server or an IP camera.

The idea behind this division is based on the fact that a processing unit already exists near each camera inside the video encoder (used to compress the video). This existing processing unit is a low-cost fixed-point processor and is highly suitable for performing several operations (as described below) that allow the sending of only a small amount of information to the image processing server for the main analysis. In this way, the system utilizes both the high resolution of the original video and the computing strength and flexibility of the central server, without the need for a costly network.

4.1 Feature extraction near the camera

The initial part of the processing, which is done by the video encoder is called the Universal Feature Extraction (UFE). This process is the part of the algorithm that works at the pixel level and extracts condensed information (or "features") from the image pixels. This process works on the incoming images when they are at their highest quality and no data has been lost due to image compression. When a suitable feature is located it is sent to the central server for further analysis over the IP network. Since the feature data is very compact, it requires a negligible amount of network bandwidth (only around 20 Kbps for each camera).

There are many types of features that can be identified in this manner, including but not limited to:

- Segmentation of foreground and background
- Motion vectors – generated by tracking areas of the image between successive frames.
- Histograms
- Specific color value range in a specified space (RGB, YUV, HSV).
- Edge information
- Identifying problems with the input video image such as image saturation, overall image noise and more.

Additionally, upon request from the server, the video encoder can send the actual pixel data for a certain portion of the image. For example, when performing automatic license plate recognition, the video encoder can send only the pixels of the license plate to the server, thus eliminating the need for more bandwidth as is the case when sending the whole picture.

The common attributes to all these features is that they can be very efficiently implemented on fixed point DSP processors on the one hand and provide excellent building blocks for a wide variety of algorithms on the other hand (hence the name Universal Feature Extractor).

***Note:** because of its low resource requirements the UFE can be implemented inside hardware that is already performing video compression without need for additional processing power. This is not the case with "local processing" based algorithms that are being ported to DSP platforms. The current generation of cost effective processors does not allow performing of high quality video encoding and high quality image processing on the same DSP. Although it is expected that future generation processors will allow this in the coming years, the "local processing" architecture will always be very limited in terms of its flexibility and support for various types of concurrent algorithms.*

4.2 Feature analysis at the central server

The main part of the processing is performed by the **IPoIP™** server. The server is able to dynamically request specific features from each camera, according to the requirements of the specific algorithms that are currently being applied.

The server analyzes the feature data that is collected from each camera, and dynamically allocates computational resources as needed. In this way the server is able to utilize large-scale system statistics to perform very complex tasks when needed, without requiring a huge and expensive network for support. The part of each algorithm that runs on the server performs the following main tasks:

1. Request specific features from the remote UFE.
2. Analyze the incoming features over time and extract meaningful "objects" from the scene.
3. Track all moving objects in the scene in terms of size, position and speed and calibrate all of this data into real world coordinates. The calibration process transforms the 2 dimensional data received from the sensors into 3 dimensional data using various calibration techniques. Many such techniques can be implemented in accordance with the specific scene being analyzed.
4. Classify these objects into one of several major classes such as vehicles, people, animals and static objects. The classification process can be done using various parameters such as size, speed and shape (pattern recognition).
5. Obtain additional information regarding objects of interest such as color, or sub classification (type of vehicle, etc.)
6. Optionally extract unique identifying features for an object, such as a license plate recognition or facial recognition.
7. Decide based on all the gathered information and on the active detection rules whether or not an event needs to be generated and the system operator informed.
8. Receive and analyze information from any other algorithm running on the server at the same time. This very powerful capability enables easy implementation of tasks such as inter-camera tracking. Using this ability a specific moving object (a person or vehicle) can be accurately tracked as it moves from the field of view of one camera to the next with the system operator always viewing the correct image. This ability also enables creating sequences of rules where a rule on one camera only becomes activated (or deactivated) when a rule on another camera detects an event.

It is important to note that the algorithms at the server are constantly gathering information regarding the scene even though most of the time no events are being generated. This information can be stored as meta-data along with the video recording and later enable very fast and efficient searches on large amounts of recorded video content.

4.3 The combined end-product

Utilizing the methods described above **IPoIP™** is able to provide algorithm complexity level and low costs that are unrivaled by any other existing method today as can be seen in the following comparison table:

Feature	Local Architecture	Server Architecture	IPoIP™ Architecture
Cost for medium to large installations	High (due to many processors)	High (due to network)	Low
Suitability for large installations	Network – Yes Hardware – No	Network – No Hardware – Yes	Network – Yes Hardware – Yes
Scalability of applications (adding new algorithms and features)	No	Yes	Yes

5 Applications in the Physical Security Market

The **IPoIP™** platform is ideally suited for applications needing multiple simultaneous image input and processing. The fastest growing market today for such large scale image processing is the Physical Security market.

Standard security measures today include the rapid deployment of hundreds of thousands of cameras in streets, airports, schools, banks, offices and residences. These cameras are currently being used mainly for enabling the surveillance of a remote location by a human operator or for recording the occurrences at a certain location for use at a later time should the need arise.

The introduction of digital video networking and other new technologies is now enabling the video surveillance industry to move to new directions that significantly enhance the functionality of such systems. As a result, video surveillance is rapidly penetrating into organizations needing security monitoring on a very large scale and in widely dispersed areas – such as railway operators, electricity and energy distributors, the Border and Coast Guards and many more.

Such organizations encounter new problems of operating and handling a huge amount of cameras, while having to provide for extensive bandwidth requirements. This is where the use of automatic video-based event detection comes into play. Solutions are currently available for automatic video motion detection (VMD), license plate recognition (LPR), facial recognition (FR), behavior recognition (BR), traffic violation detection and other image processing applications. The output of these detection systems may be used for triggering an alarm and/or initiating a video recording. This can reduce network bandwidth requirements (in situations where constant viewing and recording is not required) and allow allocation of human detection only to those cameras that contain a special event.

All the current implementations of these algorithms suffer from the inherent problems of existing system architectures as described above, and thus are very costly and unable to penetrate the market on a large scale. **IPoIP™** provides the ideal platform for a cost-effective high performance and constantly evolving physical security system.

5.1 Sample application - Railway system protection

In order to demonstrate the practical use and benefits of the IPoIP™ technology, following is a description of a typical application – **Railway system protection**. This example shares similar requirements with other applications such as borderline security, pipeline protection and more:

- **Poor infrastructure** – The power and communication infrastructure along the tracks is not guaranteed. A low-power and low-bandwidth solution is mandatory (e.g. transmitting hundreds or thousands of cameras is not practical). A wireless / solar-cell powered solution is desired.
- **Mostly outdoor environment** – The system should be immune to typical outdoor environment phenomena such as rain, snow, clouds, headlights, animals, insects, pole vibration etc.
- **Distributed locations** – Railway facilities (tracks, stations, bridges, tunnels, service depots etc.) are distributed over a large geographic area, which forces using an IP network based system.
- **Large-scale** – A typical railway system would use thousands of cameras to protect the tracks and all facilities. The (nuisance alarm ratio) NAR/FAR per channel figures should be extremely low so that the accumulative system will can effectively be monitored by a small number of operators.
- **Critical system** – The system's availability should be close to 100%. No single-point-of-failure should exist. It is desired that the network will handle local failures such as cable cuts.
- **Variety of event types** – The video intelligence system should detect intruders, suspected objects, safety hazards, suspected license plate numbers and other standard and user-specific event types. This can be achieved using multiple high level algorithms, including using several algorithms simultaneously for a single camera.
- **Low cost of ownership** – As the protected area is very large, rural and distributed, field visits are very expensive. Therefore, a minimum amount of equipment in the field is vital for low installation and maintenance costs.

Looking at the above list, it is clear that the classic concept of local processing – either field based or center based – fails to comply with most requirements. Field based solutions require lots of computers in the field, resulting high power requirements and cost of ownership. Server based solutions require transmission of all the video sources at high quality all the time to the center, resulting in very high bandwidth requirements.

Using IPoIP™ technology, only low-power video encoders with embedded feature extraction capability are required in the field. Furthermore, most of time there's no need to transmit video but only low bandwidth feature stream data, which is a dramatic saving in network bandwidth requirements without compromising on performance. Following is a description of the whole solution.

Poles are installed along the tracks. Each pole is carrying a FLIR (thermal) camera, video encoder, IP network node and power circuitry.

The FLIR camera can detect persons reliably up to few hundred meters at all weather and illumination conditions, thus preventing the need for artificial illumination and reducing FAR/NAR. The camera consumes 2-5W.

The video encoder / feature extractor unit is a low power module that uses some 10-20Kbps of feature data in average and transmits video at higher bandwidth (0.5-2Mbps) only when an event is detected or upon an operator's request. The encoder consumes 3-8W.

The IP network can be either a wired (copper or fiber) or wireless solution. For a wired network, fiber is recommended as it is not limited by distance and immune to EMI/RFI. If cabling is not possible or is too expensive, a wireless solution may be used. A hybrid WI-FI and satellite based network is recommended such that the inter-pole communication is WI-FI based and the access points use satellite link. An antenna should be installed on the top of the pole. This solution does not require any infrastructure and consumes about 10W per pole / 40W per access point.

Power may be supplied either by power lines or using a solar cell and battery module. If cabling is used, it makes sense to use power lines. If a wireless network is used, the power should be supplied by solar cells.

On top of the FLIR cameras used for intruder detection, a PTZ color camera is installed every 2-4 Km for event monitoring and management.

Two algorithms are used to protect the railroad. A Video Motion Detection (VMD) algorithm is used to detect persons and vehicles approaching the protected area. A Non-Motion Detection (NMD) algorithm is used to detect any static changes in the scene such as objects left on tracks (bomb, fallen tree, stuck car) or damaged tracks (missing parts). These two algorithms are used simultaneously.

The server is located at the backend and is based on a cluster of two or more computers designed as required for critical systems, i.e. no single point of failure, full redundancy and backup. The server computers may even be geographically distributed over few locations to increase robustness.

The system may be operated from any location on the network. This enables dividing large networks to various users / departments.

For further information please contact:

Agent Vi Inc.
Global Offices
245 Park Avenue, 39th Floor
New York, NY 10167
+1 212-672-1620

www.agentvi.com
info@agentvi.com