REAL-TIME THERMAL CUEING SIMULATION: A PHYSICAL-BASED APPROACH

Romolo Gordini

Abstract

The targeting/laser POD Thermal Cueing simulation is a fundamental component of Full Mission flight simulators for training military pilots. It combines the capabilities of Thermal Imaging and magnified visible light (Electro-Optical), including high definition mid-wave forward looking infrared (FLIR) imaging.

Pilots can select the best POD imaging mode (e.g. FLIR mode or RECCE mode) and, depending on the specific nature of the mission, activate Thermal Cueing. This significantly increases the combat effectiveness of the aircraft in all weather conditions, for both the attack of ground and air targets, effective with a large variety of standoff weapons.

Once a target has been located on an imaging mode via Thermal Cueing, the laser designator can be used to guide a laser guided bomb towards the reflected laser energy.

Thermal Cueing training solutions deployed in simulation normally use, in order to locate hot/white spots on POD images, either synthetic information coming from the Synthetic Environment (SE) or delegate the Thermal Cueing chevron designations to the Visual System. SE uses synthetic information which is a-priori known by the simulation and does not take into account in any respect either the actual POD image nor any physics. On the contrary, the Visual System considers the POD images purely, with a poor, if any, simulation of physical properties of materials in different frequency bands, nor consider in any respect the detection capability of the device itself nor environmental conditions.

These two different approaches are the only existing methods capable of guaranteeing basic Thermal Cueing simulation with the time-critical data I/O operations with the rest of the simulator but they dramatically fail in assuring sensor realism to a high degree of fidelity.

This paper describes the initial results of Leonardo research using a prototype real-time Thermal Cueing simulation capable of analysing in real-time images coming from sensor cameras (one or more Image Generator channel in any frequency band). It performs the extractions from the input video of the most significant targets according to physical rules (hot/white spots), collapse the candidate points passing the filtering phase into a smaller number of significant clusters and track them among consecutive frames of the video flow.

The key point of this research was to tailor complex video processing algorithms into a physical-based simulation software able to run in real-time on commercial hardware. The model guarantees physical realism and allows the direct use of the true POD data package for its Thermal Cueing customization. Results permit to combine high fidelity computation with excellent realism.

The fidelity of the resulting software was validated by comparing the simulator Thermal Cueing results with the outputs from military videos.

Further research into adjacent fields, such as on real POD test beds in order to early diagnose possible problems and thus drastically reduce the use of expensive test campaigns in operational scenarios. Possible other dual usages of this technology are the monitoring of public events and medical research, using micro cameras to diagnose diseases at an early stage of development.

GENERAL DESCRIPTION OF THE REAL EQUIPMENT

Targeting/laser PODs mounted on military interceptors, optimized for the air superiority role in beyond visual range and close air combat, normally provide multi sensors, laser designator/range finder systems for target detection, recognition and weapon delivery. These pods can operate both in Air-to-Surface and in Air-to-Air modes, allowing the acquisition and tracking of air and ground targets.
Such sensors, taking advantage of latest image processing algorithms, combined with advanced stabilization techniques provide cutting edge performance. High-resolution mid-wave IR and EO sensors operate in conjunction with a laser mode, providing eye-safe theatre operations and precise geo-location features.
Pilots can select the best POD imaging mode, depending on the specific nature of the mission, and activate Thermal Cueing. This significantly increases the combat effectiveness of the aircraft in all weather conditions during the attack of ground and air targets with a large variety of standoff weapons.

Once a target has been located via Thermal Cueing, the laser designator can be used to guide a laser guided bomb towards the reflected laser energy.

PODs purpose normally is:

- detection, acquisition, tracking and imaging of possible targets and threats in the air-to-air operation;
- detection of possible ground targets and threats in the flying aid mode in all weather;
- provision of flying and landing aids for the pilot at night, and in adverse weather conditions.

The current paper is focused on Targeting/laser PODs Imaging Modes in IR and its implementation into Full Mission flight simulators for training military pilots.

TARGETING/LASER POD REAL-TIME SIMULATION, STATUS

Within a training system, the Targeting/laser POD real-time simulation (typically at 50 Hz) has got a dual nature. It is primarily a sensor, but it is also part of the Ownship Weapon System Simulation. Two architectural solutions are therefore common. The former (more consolidated and transitional) hosts the POD LRI within the Weapon System Simulation Host, the latter (more innovative and flexible) hosts the POD LRI within the Sensors Host.
There are positive and negative aspects for both architectural approaches, whose examination goes beyond the purpose of this paper. In both cases, PODs interface with the simulation and can be summarized as follows.

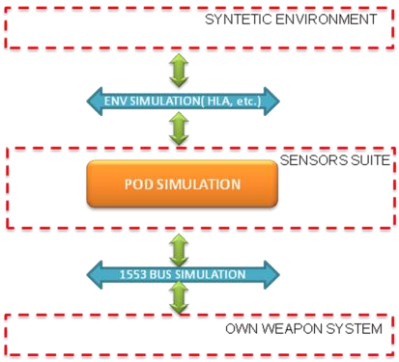

Figure 1 – POD logical interfaces.

There are two major logical interfaces to the POD, one via the 1553 Bus, either simulated or real and the other via commercial protocols, typically DIS, HLA, TCP, RMB or other ones with the Synthetic Environment.
The POD System's dual nature dictates for it the following simulation requirements:

1. to be highly correlated with other mission data sets from other weapons sensors, such as radar, IFF, MIDS, etc. and synthetized into a single target information set by the Attack Computer;
2. to be highly correlated with other flight data sets from other mission sensors, such as Visual System, Navigation System, etc. and perceivable by the Pilot through his eyes;
3. to faithfully represent the equipment it is simulating, with all modes, data and performances as design by the equipment OEM.

These needs are frequently conflicting with one another, especially for IR Thermal Cueing.
In fact, Thermal Cueing training solutions deployed in simulation – in order to locate hot/white spots on POD images – either use synthetic information coming from the Synthetic Environment (SE), or delegate the Thermal Cueing chevron designations to the Visual System.
SE uses synthetic information which is *a-priori* known by the simulation and does not take into account in any respect either the actual POD image nor any physics.
On the contrary, the Visual System considers the POD images purely, with a poor, if any, simulation of physical properties of materials in different frequency bands, nor consider in any respect the detection capability of the device itself nor environmental conditions.
These different existing approaches are the only ones capable to guarantee Thermal Cueing basic simulation with the time-critical data I/O operations with the rest of the simulator but they dramatically fail in assuring sensor realism.

THERMAL CUEING REAL-TIME SIMULATION

This paper describes the Leonardo *in-house* TC research using a prototype real-time Thermal Cueing simulation capable of analysing in real-time images coming from sensor cameras (one or more Image Generator channel in any frequency band), which perform the extractions of the most significant targets according to physical rules (hot/white spots), collapse the points passing the filtering phase into a smaller number of significant clusters and track them among consecutive frames of the video flow.

Since the beginning the architectural schema used to implement the solution is original. The simulated sensor, in fact, does not have a proprietary IG resident on the Sensor Suite Server, as for instance in the EF2000 ASTA solution, but it starts from the concept that the unique IG is the OTW one and that the correlation with it is a must have bonus. Up to now this approach has been generally discarded because it implies acquiring in real-time, via commercial frame grabbers/acquisition boards, the stream of images and processing them on the fly, with physical algorithms of image processing. This method is very accurate and precise, but it normally requires a significant amount of time, and up to now, it was not considered possible to tailor it in order to be used with a processing rate compatible with the one adopted by the simulation.

The functional block diagram of the sensor is the following:

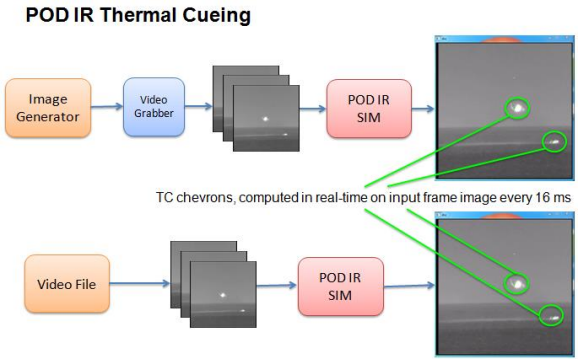

Figure 2 – POD functional blocks.

The architectural block diagram is the following:

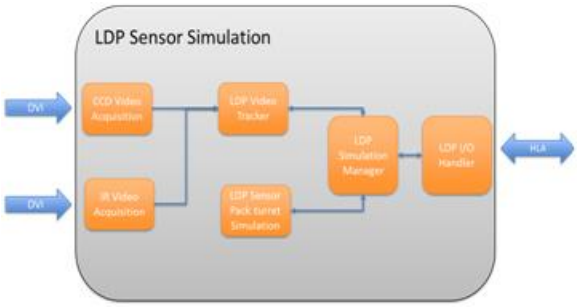

Figure 3 – POD architectural blocks.

The core of this model is the Video Tracker. Its main characteristics are:

- Real Time processing of a sequence of images to perform the Thermal Cueing functionality;
- Black Hot / White Hot TC;
- Independent from the sources of video (video files, Image Generator, etc.);
- Multithreaded C++ or GPU programming;
- Output data are digital TC data.

VIDEO TRACKER ALGORITHM DESCRIPTION

The input image is acquired in real-time via an acquisition board and it is decomposed into a sequence of images which are in turn stored into a queue ready to be processed by the SW.

Pre-processing

For each image, the algorithm performs a Pre-processing Phase, which consists of two separate steps:

1. Gaussian smoothing, in order to reduce the noise (image on the left);
2. Threshold-based segmentation, in order to speed-up execution (image on the right).

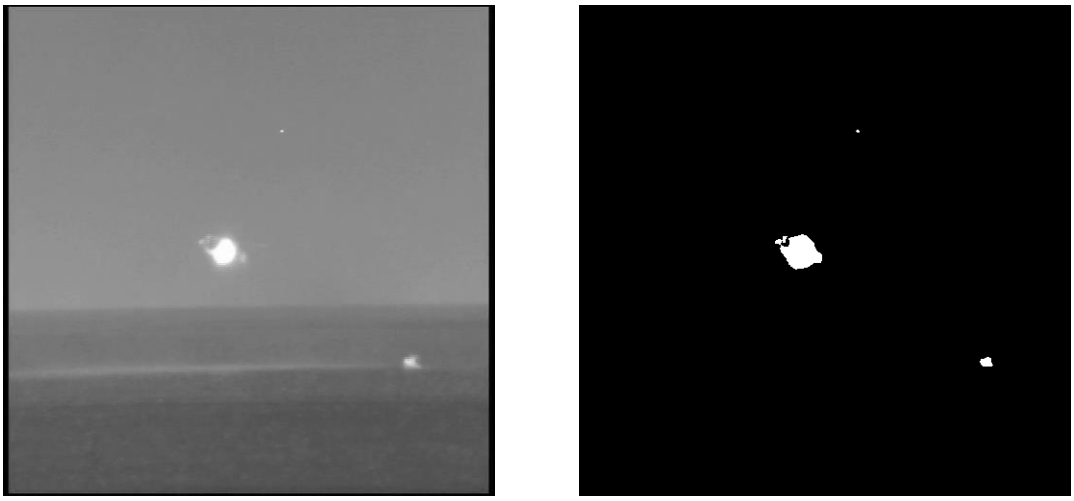

Figure 4 – Input image, pre-processing phase.

The following phase, "Target Detection & Tracking", uses a worldwide known algorithm (KLT) to extract targets from an image. The KLT algorithm has been modified in order to be adapted to strict real-time processing.

The KLT algorithm has logically got two separate and distinct sub-phases, further tailored for Leonardo requirements:

- *Target Detections*: on the first frame of a sequence of images it extracts "outstanding points" (corners);
- *Target Tracking*: it estimates on the next frame of the sequence of images locations of targets detected into the previous phase (sequential mode). A Kalman filter has been added to the logic, in order to cope with situations of tracks fading or cycles of lack of detections.

Target Detection

The basic concept of this phase is to analyse the entire starting image, looking for a corner, by using a sliding squared window of known dimensions and centring it on each pixel at each iteration.

$$C_{str} = \begin{bmatrix} c_{xx} & c_{xy} \\ c_{xy} & c_{yy} \end{bmatrix}$$

on (x,y) pixels marked as white in mask

Figure 5 – Input image, target detection.

The window identifies the portion of the image to analyse to determine if the pixel on which it is centred is a corner and then a target. Having extracted this portion of the image, the following step – in order to determine if in it there is a corner – is to analyse the gradient.

For this reason, prior of the target selection process, two matrices of gradients, namely "$grad_x$" and "$grad_y$", have to be calculated. The dimensions of these latter matrices are equal to the dimensions of the initial image. They respectively contain the gradient along the x axis and the gradient along the y axis of the light intensity of the initial image (previously smoothed).

At this point the same "*selection window*" has to be picked up from gradx and grady matrices, centred on the pixel that the algorithm wants to determine whether it is a target or not, and consider the values of these two extracted windows which have to be combined in order to obtain the 2x2 G array, whose elements are: g_{xx}, g_{xy}, $g_{yx} = g_{xy}$, g_{yy}.

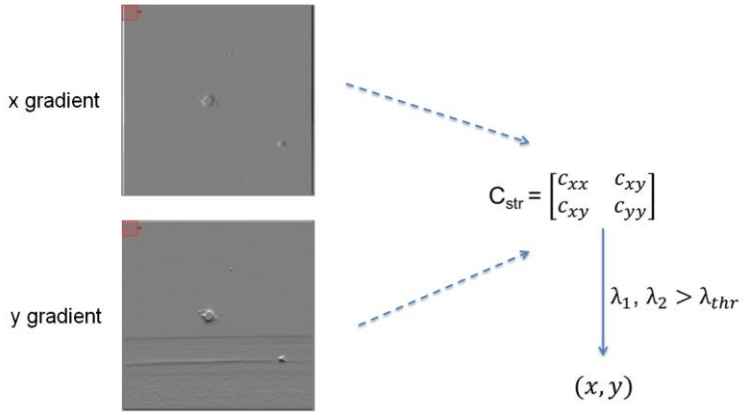

Figure 6 – Gradient images, structure matrix.

The matrix G is called "*Structure Matrix*", since its eigenvalues define the presence or absence of edge or corner in the pixel under evaluation. In particular:

- g_{xx} is the sum of all the squared values of the pixels contained in the window extracted by gradx;
- g_{yy} is the sum of all the squared values of the pixels contained in the window extracted by grady;
- g_{xy} is the sum of all the values of the pixels contained in the window extracted by $grad_x$ multiplied by the corresponding values of the pixel matrix extracted from $grad_y$.

The pixels for which the G matrix has got a non-zero determinant are "*target candidates*".

In general, the more the values of elements of the matrix G are smaller (if close to zero, the determinant will become zero), the more probable is the fact that the window under evaluation does not contain variations in light intensity and therefore does not contain corners.

A candidate target becomes effectively a corner point if the smaller eigenvalue of the G matrix exceeds a certain lambda threshold.

After analysing the entire input image with this sliding window, which scrolls centred on each pixel and evaluates the matrix G for each pixel, a list of corner points (x, y) can be defined and for each of them, the relative value of the smallest value of the associated G matrix.

The list is sorted by decreasing eigenvalue values. At this point the distance between the corner points in the list shall be evaluated to select only one point for the same area (it is highly probable that more pixels are returned as targets for corners of an object into the image). This point will be the one with the highest eigenvalue value among all those contained in a 2D x 2D window, where D is the parameter which considers the minimum settable distance between two possible corners.

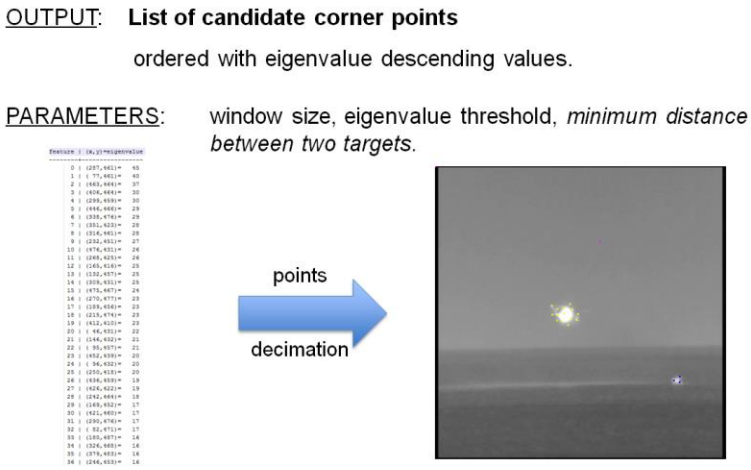

Figure 7 – Input image, eigenvalues.

Target Tracking

The tracking algorithm takes the input list of targets obtained from the previous step and derives the estimated position (*x* + *dx, y* + *dy*) for each target in the next image of the sequence.

The algorithm has been heavily customized in order to optimize steps to be performed at each real-time iteration.

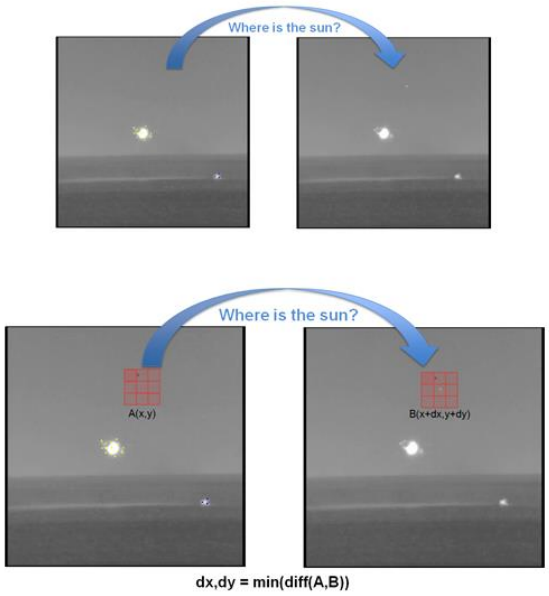

Figure 8 – Input image, target tracking.

The computation generates a pyramid of images for each frame of the image stream, in which each layer represents the original image, with a resolution reduced by a factor of 2 compared to the image of the previous level.

In this way the algorithm can follow targets with more than one level of results, starting from the lowest resolution layer of the image.

Increasing the number of levels of the pyramid allows the algorithm to handle cases in which the offset between corresponding targets in successive frames is large, at the cost of an increase in

the computational load. The number of levels of a pyramid is calculated when creating the tracking context during the pre-processing phase.

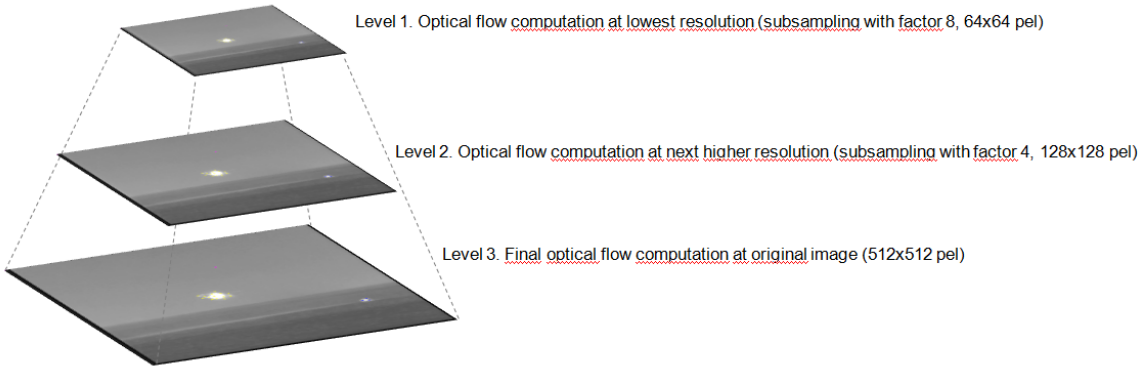

Figure 9 – Input image, pyramid computation.

Each level of the pyramid is obtained through sampling the previous level and applying a factor of 2 in width and height. The tracking algorithm starts from tracking targets in the lower resolution level and continues up to convergence. The result obtained on one level is propagated to the next level as an initial guess for the position of the target in the frame. In this way, the tracking is refined in each level more and more, up to the original image. The use of pyramid levels allows the tracker to handle larger movements of the neighbourhood size.

Post-processing

After the detection and tracking steps of the algorithm, for each object with a “thermal outstanding signature” into the image frames targets are extracted and estimated, see figure below.

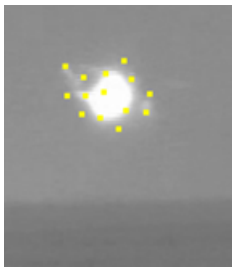

Figure 10 – Input image, post-processing phase.

The final goal is to have only one target per object for such cases. A post-processing phase on the image is therefore needed. For this step the DBSCAN algorithm has been selected, again, suitably modified in order to sustain hard real-time processing.

DBSCAN stands for Density-Based Spatial Clustering of Applications with Noise. Its major characteristics are:

- an arbitrary number of clusters with arbitrary shape;
- the capability to recognize noise.

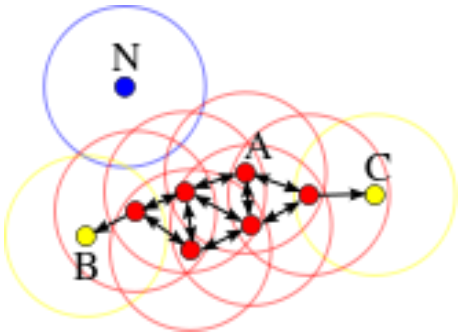

Figure 11 – Clustering.

For this purpose the concept of cluster has been introduced. A Cluster is a set of mutually *density-connected* points (border points are only *density-connected*).

Points *p* and *q* are *density-connected* if it exists point *o* that is *density-reachable* from *p* and *q*.

Two points *p* and *q* are *density-reachable* if it exists a sequence of points $[p_1, \ldots, p_n]$, with $p_1 = p$ and $p_n = q$, *s.t.* p_{i+1} is *directly reachable* from p_i.

Point *q* is directly reachable from point p if:

- *dist (p, q)* < ε, and
- *p* has at least minPts points in its neighbourhood.

After this phase the image processing is complete. In this example two targets are identified and tracked, with the sea and the sun both being considered as noise.

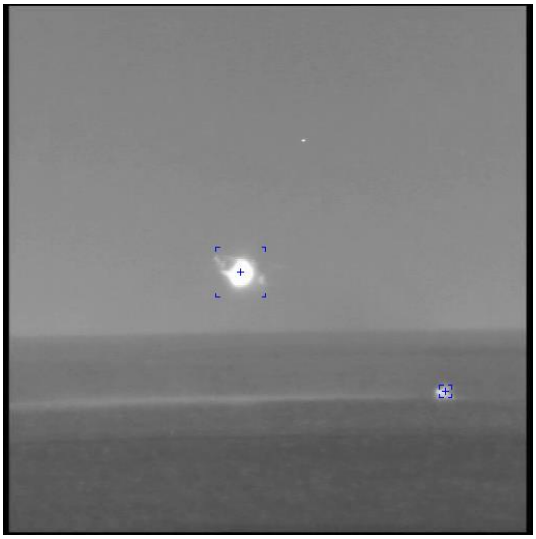

Figure 12 – Output image, final result.

FUTURE ACTIVITIES

The model could be used to represent cost savings in adjacent fields, such as on real POD test beds in order to early diagnose possible problems in the equipment development process and thus drastically reduce the use of expensive test campaigns in operational scenarios. Possible other dual usages of this technology are the monitoring of public events and sensitive targets surveillance, via the detection and tracking of potential thermal threats using video streams from cameras.

This technology could finally be used in medical research, using micro cameras to diagnose diseases at an early stage of development.

CONCLUSIONS

The key point of the work was to tailor complex and consolidated video processing algorithms into a physical-based simulation software able to run in real-time on commercial hardware. The model guarantees physical realism and allows the direct use of the true POD data package for its Thermal Cueing customization. Results permit to combine high fidelity computation with excellent realism.

The fidelity of the resulting software was validated by comparing the simulator Thermal Cueing results with the outputs from military videos coming from IRST flight trails.

BIBLIOGRAPHY

[1] **Kanade-Lucas-Tomasi (KLT) Feature Tracker,** Computer Vision Lab. Jae Kyu Suhr, Computer Vision (EEE6503) Fall 2009, Yonsei Univ.

[2] **A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise**, M. Ester, H. P. Kriegel, J. Sander, X. Xu, *Institute for Computer Science, University of Munich, Oettingenstr. 67, D-80538 Munchen, Germany*, KDD-96 Proceedings 1996, AAAI.

[3] **MARS (Multimode Airborne Radar Simulator) User's Guide***, S. Cavallaro, R. Gordini,* Leonardo Spa*,* 2011.